Past Examinations for Theory of Algorithms

by

Peter M. Maurer
1. Prove that for all \( n \), \( x^{n+1} \) grows faster than \( x^n \).  
(10 points)

2. Prove that \( cf(n) \in \Theta(f(n)) \).  
(10 points)

3. You need to sort a collection of 100-character items, the whole item is to be used as the key. Assume that string comparisons take the same amount of time as character comparisons. You are given the choice between a heapsort algorithm and a radix sort algorithm.
   a. How would the size of the file influence your choice of algorithm?  
      (5 points)
   b. At what point (specifically) would you switch from one algorithm to the other?  
      (10 points)
   c. How much storage would radix sort take at the switch point, assuming that the keys contain only upper case letters and spaces?  
      (5 points)

4. Solve the following recurrence assuming that \( T(0)=0 \) and \( n \) is a power of 2.  
(10 points if your correct solution contains a summation.)  
(5 additional points for removing the summation.)
\[
T(n) = cn + T(n/2)
\]

5. How many comparisons would be required in the worst case to find the largest, second largest, and third largest keys in a collection of unordered items?  
(10 points)

6. Find the minimum spanning tree of the following graph, and the shortest path from node X to node Y.  
(10 points for MST, 5 points for shortest path.)

![Graph Image]
1. List all the articulation points (5 points) and the biconnected components (5 points) of the following graph.

2. Using the graph given in the previous problem, show the order that the nodes would be visited by a depth first search (5 points) and a breadth first search (5 points).

3. Find the strongly connected components of the following directed graph (5 points) and the condensed graph (5 points).

4. Using Strassen's method of matrix multiplication, how many additions and multiplications would it take to multiply two 8x8 matrices? How many multiplications and additions would be required using the standard algorithm? (Answers of the form $\Theta(n^3)$ are not acceptable.) (10 points).

5. Factor the following polynomial using Horner's rule. (10 points).

\[ 3x^5 + 7x^4 + 2x^3 + 8x^2 + 4x + 9 \]
6. Given the following object sizes, .7, .6, .5, .25, .25, .2, .2, .15, .15, give an optimal bin packing for these objects (5 points) show how the Niff algorithm would place these objects in the various bins (5 points).

7. What is the maximum number of colors required to color an n-vertex graph? What is the minimum number of colors to color an n-vertex connected graph? For n=5 give examples of a graph that requires the maximum number of colors, and a connected graph that requires the minimum number of colors. (10 points)

8. Suppose a friend of yours is trying to prove that P=NP using the following technique. Suppose you are given a logical expression which is an input to SAT. First determine the length of the expression, suppose the length is n. Now for every variable, x in the expression replace x with xaaa...aaa so that there are $2^n$ a's at the end of the original name. Now the length of the expression is greater than $2^n$. Since you now have an expression with no more than n variables which is at least $2^n$ characters long, you can determine whether the expression is satisfiable in $O(n)$ time by doing a brute force search of all possible inputs. This proof does not work. Why not? (10 points)
1. Find the function $W(n)$ that describes how many X operations the following algorithm does for an input $n$. (10 points for the recurrence relation, 10 points for solving the recurrence.)

```plaintext
procedure abc(n)
    for i = 1 to c do
        for j = 1 to n do
            X-operation
        endfor
    endfor
    if (n>=2)
        abc(n-2)
    endif
endprocedure
```

2. As we all know, Radix sort is order $n$, Quicksort is order $n^\log_2 n$, and Insertion sort is of order $n^2$. Nevertheless, both Quicksort and Insertion sort have been found to be significantly faster than Radix sort. Why? (20 points)

3. How many comparisons would it take to find the largest, smallest, second largest and second smallest keys in an unordered list of $n$ objects. Explain your answer. (20 points)

4. Prove that for all integers $n>0$ and all $\alpha>0$, $x^n+\alpha$ grows faster than $x^n$. (10 points)

5. Prove that for all constants $c$, $f(n)+c\in \Theta(f(n))$. (10 points)
1. For the following graph in what order would the vertices be during a depth first search? During a breadth first search? (Start with vertex A.) (20 points).

2. Find the minimum spanning tree of the following graph, and the shortest path from A to E. If there is more than one minimum spanning tree, show them all. If there is only one, explain why there is only one. (20 points).

3. Find the biconnected components of the following graph. (15 points).

4. Find the strongly connected components of the following directed graph. (15 points).
5. In Strassen's matrix multiply, suppose the operations to perform a 2×2 matrix multiplication took 8 multiplies instead of 7. How would this affect the time bound? (You may restrict your answer to powers of 2). (15 points).

6. Fast Fourier Transform gives us a fast way to evaluate a polynomial several times on different values of x, in particular for the values $x^0, x^1, ... x^{n-1}$. Why won't the procedure work if x is not a primitive root of unity? (15 points).
1. (20 points) A friend of yours thinks he knows how to prove that P=NP by demonstrating that the satisfiability problem can be solved in polynomial time. This is his argument. "First consider the problem of determining whether an equation in DISJUNCTIVE normal form is satisfiable. (In the form \( ab+cd' \) for example.) We look at every term, if a term contains both a and a', that term is always zero. On the other hand if a term DOES NOT contain both a variable and its complement, then the term takes the value 1 for some set of inputs. We can test every term for this condition in \( n^2 \) time. So first we transform our equation into DISJUNCTIVE normal form and then test it. QED." What is wrong with this argument?

2. (20 points) What does it mean when one says that a problem is NP-Complete? BE SPECIFIC!

3. (20 points) Prove that if the degree of all vertices of a graph is limited to 2 or less, that the sequential coloring algorithm always uses the minimum number of colors.

4. (10 points) What is the minimum time bound for computing an arbitrary boolean function of \( n \) variables using \( n \) processors? On page 371 we find a constant time algorithm for the inclusive OR function of \( n \) bits. Does this imply that section 10.5 is wrong?

5. (10 points) True or untrue: Using \( n \) processors, the fastest I could possibly sort a list of \( n \) numbers is \( \lg n \). Why?

6. (20 points) Give the KMP fail indexes and the Boyer-Moore matchJump array for the following strings.

   AAAB
   AABACAABABA
   ABRACADABRA
   ASTRACASTRA

7. (extra credit, 25 points) Suppose the bin packing problem is restricted so that the number of objects of different sizes is bounded above by some constant. Is the problem still NP-complete? Why or why not?
1. Prove that for all \( n \) \( x^n \lg(n) \) grows faster than \( x^n \).
   (15 points)

2. Prove that \( cf(n) \in \Theta(f(n)) \) for all \( c > 0 \).
   (15 points)

3. Solve the following recurrence assuming that \( T(1)=0 \) and \( n \) is a power of 2.
   (10 points if your solution contains the correct summation.)
   (5 additional points for removing the summation.)
   \[
   T(n) = n\lg(n) + T(n/2)
   \]
   (hint: after obtaining the summation, write \( n \) as a power of 2, and set \( i=\lg(n) \).)

4. Give an optimal algorithm for finding the maximum and minimum of four numbers.
   (15 points)

5. Give an algorithm for sorting four keys that uses only five comparisons in the worst case. (Hint, use the solution to problem 4.)
   (15 points)

6. Suppose you are given an algorithm for finding the median of a list of \( n \) numbers using \( cn \) comparisons in the worst case \( (c > 0) \). Assume that this algorithm is implemented as a function MEDIAN(L) which returns the index of the median in the list L. What is the worst-case running time of QUICKSORT assuming you use this function in SPLIT to find the median? Justify your answer.
   (15 points)

7. Recall the implementation of a binary tree used in HEAPSORT. Suppose we have a different sort of algorithm that requires the use of a complete TRINARY tree. In such a tree, every non-leaf has exactly 3 children, and all leaves are on the same level. Assume that this tree is mapped into an array in the HEAPSORT fashion. Assuming that the index of a node is \( i \), give a formula for the left, middle and right children of the node. (Work this problem LAST.)
   (10 points)

Food for thought: (NOT PART OF THE EXAM) could HEAPSORT be implemented using TRINARY trees? If so, what would the worst-case time-bound be?
1. For the following graph in what order would the vertices be during a depth first search? During a breadth first search? (Start with vertex A.) (15 points).

![Graph](image1)

2. Find the minimum spanning tree of the following graph, and the shortest path from H to Q. If there is more than one minimum spanning tree, show them all. If there is only one, explain why there is only one. (20 points).

![Graph](image2)
3. Find the biconnected components of the following graph. (15 points).

4. Find the strongly connected components of the following directed graph. (15 points).

5. \( T(0) = 1 \)
\( T(n) = 3n + 2T(n/2) \)
Find \( T(2^{32}) \) and the general form of \( T(n) \). (20 points).

6. How many multiplications would it take to evaluate this polynomial blindly using Horner's rule?
What is the minimum number of multiplications for this specific polynomial? (15 points).

\[ 32x^{32} + 16x^{16} + 8x^8 + 4x^4 + 2x^2 + x + 1 \]
1. (20 points) While performing a Fast Fourier Transform, we compute the value of $n$ polynomials of degree $n$ in $n \lg(n)$ time. However, the minimum time bound for computing polynomials leads us to believe that the minimum time bound for computing the value of $n$ polynomials of degree $n$ is $n^2$. Can we use the FFT technique to compute the value of $n$ polynomials of degree $n$ in $n \lg(n)$ time? If not why not? If so, then what is wrong with the minimum time bound we found for polynomial evaluation?

2. (20 points) In Strassen's matrix-multiply, if it were possible to reduce the number of multiplications from 7 to 4 without changing the number of additions, how would it affect the time bounds for addition and multiplication? This is your recurrence problem, so be specific. (Hint: this problem is easier than it first appears.)

3. (20 points) Suppose a friend of yours is trying to prove that P=NP using the following technique. Suppose you are given a graph and you want to determine if it is 3-colorable. You can determine if it is 3-colorable by trying all $3^n$ combinations of 3 colors. So you add $3^n$ vertices to the graph and make sure that these are isolated vertices that are not connected to any other vertex. Since they are isolated, they do not change the number of colors needed to color the graph. In fact we can just assign them all the color 1 and forget about them. Since the graph now has more vertices than combinations, we can do the brute-force search in polynomial time. There are two reasons why this proof does not work. What are they?

4. (20 points) Consider the bin-packing problem with the following limitation: the number of different sizes of objects is always limited to 3. (The specific sizes are not known.) Is this problem NP-complete? If so prove it. If not show why not.

5. (20 points) Given a polynomial $p$ and a value $x$, devise a PRAM algorithm for computing $p(x)$. Is your algorithm optimal? What is the optimal time-bound for this problem?
1. Prove that for any integer c>0 and real number c>0, n^{k+c} grows faster than n^k.
   (15 points)

2. Solve the following recurrence relation.
   (20 points)
   \[ S(n) = n^2 + S(n-1) \]
   \[ S(1)=1 \]

3. Solve the following recurrence relation. (First figure out what values of n the function T is defined for.)
   (20 points)
   \[ T(n) = n^2 \log n + 4T(n/2) \]
   \[ T(1)=0 \]

4. Suppose that one could merge two lists of sizes m and n using only (m+n-1)/2 comparisons. How would this affect the time-bound of mergesort?
   (15 points)

5. The following algorithm sorts a list L using the "Repeated Maximum" algorithm. What is the time-bound of this algorithm?
   (15 points)
   
   for i := 1 to n-1 do
     for j := i+1 to n do
       if (L[j] > L[i]) then
         temp := L[i];
         L[i] := L[j];
         L[j] := temp;
       endif
     endfor
   endfor

6. Suppose a theatre has 500 seats which are numbered from 1 to 500. On one particular night in June the theatre presented a performance by Luciano Pavarotti which was sold out. For reasons that are too complicated to go into here, the theatre wished to print a list, by seat number, of those individuals who attended the concert. As individuals entered the theatre, the door attendants took each ticket and entered the seat number and the name of the individual into a computer terminal. Each such entry created one record. How many key-to-key comparisons would it take to sort these records into ascending order by seat number in the worst case? What is the time bound of the optimal algorithm for sorting these records? (You are permitted to be clever.)
   (15 points)
1. How many comparisons would it take to find the largest, second largest, smallest and second smallest integers in an unordered list? (This means MINIMUM folks.) Give your algorithm. (15 points)

2. Solve the following recurrence relations. What is the domain of F? (20 points)

   \[ F(n) = n+2 + 2F(n/2) \]
   \[ F(1) = 3 \]

   \[ F(n) = n+2 + 2F(n/2) \]
   \[ F(1) = 0 \]

3. Please refer to page 171 of the text. We want to construct an algorithm for finding the LONGEST simple (no vertex appears twice) path between two given vertices. We do this by altering the shortest path algorithm on pages 171 & 172 in the following way. We change the 5th line from the bottom of page 171 to read:
   
   if status[y] = fringe and dist[x] + \text{ptr}^{\uparrow}.weight > dist[y] then
   
   and the 13th line from the top of page 172 to read:
   
   traverse the fringe list to find a vertex with maximum dist;

   Does this algorithm work? If so, prove it. If not, give a counter-example and explain why not. (20 points)

4. Show that depth-first search, breadth-first search, and topological-order search are different by exhibiting digraphs for which the algorithms will visit the vertices in different orders. (Consider the algorithms in pairs.) (15 points)

5. For the biconnected component algorithm, give an example of a graph in which there are two vertices, v, w, such that back[v] = back[w], but v and w are in the same biconnected component. Do the same for the strongly connected component algorithm. (Except in this case, of course, low[v] = low[w].) (15 points)

6. Find all augmenting paths in the following flow graph. What is the maximum slack of each path? What is the maximum flow from s to t? (15 points)
1. (20 points) Solve the following recurrence relation.

\[ T = T(n/2) + 3n \]
\[ T(1) = 3 \]

2. (20 points) a. There are two things you must prove to show that a problem is NP-Complete. What are they?

   b. When transforming a known NP-complete problem such as SAT (SAT is, of course, a set of expressions, e) into a new problem which we will call ABC (a set of graphs, g), there are three things we need to prove about the transformation T, regarding both its overall properties and its effect on the elements of SAT. What are they?

3. (20 points) Consider the set of problems SUBSET-SUM-k where the number of elements in the input is restricted to k. Are these problems NP-Complete for each k? Justify your answer. (Recall that the approximation algorithm covered in class actually solved the SUBSET-SUM problem, not the KNAPSACK problem.)

4. (20 points) Give a CREW-PRAM algorithm for computing the product of n numbers.

5. (20 points) How many operations would it take to evaluate n independent nth-degree polynomials on n independent inputs? Why can FFT evaluate n polynomials faster than this?

Food for thought: (No points but fame and fortune if you come up with the right answer and prove it!) Suppose your fairy godmother gave you a magic box that you could use to instantaneously answer the SAT question: “Is expression e satisfiable?” You can stick this box in your computer and use it as a subroutine in a program. Furthermore, assume that you can use this box as a subroutine in both deterministic and nondeterministic programs. Now is P(using the box) = NP(using the box)? That is, is there a problem that can be solved in polynomial time by a nondeterministic algorithm using the box that cannot be solved in polynomial time by a deterministic algorithm using the box? The answer to this question is unknown. See Stockmeyer, L., “The Polynomial Time Hierarchy,” *Theoretical Computer Science*, Vol 3, 1976, pp. 1-22 for more information.
1. Prove the following formula by induction.
   (Remember that \((n + 1)^3 = n^3 + 3n^2 + 3n + 1\)
   (15 points).
\[
\sum_{i=0}^{n} i^2 = \frac{2n^3 + 3n^2 + n}{6}
\]

2. Prove that for any real number \(c > 1\) \(2^{cn}\) grows faster than \(2^n\). (Hint: Remember the limit formulation.)
   (15 points)

3. How many times does the function ABC print “HELLO WORLD” for an argument \(n^30\)? Find a closed form function. (Hint, first find the recurrence relation, and solve it exactly.)
   (20 points)
   
   ```
   ABC(n)
   if (n > 0) then
       for i := 1 to n do
           for j := 1 to n do
               PRINT “HELLO WORLD”;
           endfor
       endfor
       ABC(n-1);
   endif
   ```

4. Find the order of \(T(n)\) for the following recurrence relation, using the master method.
   (20 points)
\[
T(n) = 27 T(\frac{n}{3}) + n^2 \sqrt{n}
\]

5. Suppose the merge operation of merge sort were made twice as fast. How would that affect the exact solution of the recurrence specifying the time bound of merge sort. Assume that the original recurrence is \(T(n) = 2T(n/2) + n\), \(T(1) = 0\). (Hint: it is NOT necessary to solve the recurrence.)
   (15 points)
6. Order the following functions by growth rate, low to high. If two functions grow at the same rate, indicate that fact in some clear way.
(15 points)

a. \( n \)
b. \( 2^n \)
c. 17
d. \( \log_2(\log_2 n) \)
e. \( \log_2 n \)
f. \( n^4 + 4n^3 + 6n^2 + 4n + 1 \)
g. \( n^4 \)
h. \( n^3 + 3n^2 + 3n + 1 \)
i. \( n^3 \)
j. \( n! \)
\[
\begin{cases} 
  n^3 & (0 < n \leq 10,000) \\
  n^2 & (10,000 < n \leq 1,000,000) \\
  n & (n > 1,000,000)
\end{cases}
\]
k. \( n^{-1} \)
l. \( 256n^2 \)
m. \( n^{1-\varepsilon} \) (\( 0 < \varepsilon < 1 \))
1. As you already know, heaps are based on binary trees. However, let us suppose that we wanted to use TRINARY trees instead, and we want to put them into an array whose initial index is 1. Given a vertex at array position $i$, how do I find the 3 children of $i$? How do I find the parent of $i$? How would trinary trees affect the time bound of heap sort, asymptotically? Exactly? PROVE YOUR RESULT!
(20 points, 1 point for the correct answer, 19 points for the correct proof).

2. We already know that Quicksort is $\Theta(n^2)$ if the pivot point is always chosen to be the smallest element in the unsorted list, and $\Theta(n \log n)$ if the pivot point is always chosen to be the median. What is the asymptotic running time of Quicksort if the pivot point is always chosen to be the 2nd smallest element? PROVE YOUR RESULT!
(20 points, 1 point for the correct answer, 19 points for the correct proof).

3. In Dijkstra’s shortest path algorithm, when a vertex $v$ is relaxed, the distance assigned to $v$ is guaranteed to be the shortest distance from the source vertex to $v$. Show by counterexample that this is not true for the Bellman/Ford algorithm, even when the distance assigned to $v$ is less than infinity.
(20 points)

4. Let $G=(V,E)$ be a graph with vertex set $V$, and edge set $E$. Let $T=(V,E')$ ($E' \subseteq E$) be a minimum spanning tree of $G$, and let $\{u,v\}$ be an edge of weight $w$ such that $\{u,v\} \notin E'$ (not part of the spanning tree), but there exists an edge $\{x,y\} \in E'$ of weight $w' \leq w$. Prove the following.
   a. If $\{u,v\}$ is added to $T$, creating $T'$, $\{u,v\}$ will be part of a cycle in $T'$.
   b. In $T'$, none of the edges in the cycle containing $\{u,v\}$ has a weight greater than $w$.
   c. If in $T'$, there is an edge $\{x,y\} \cap \{u,v\}$ of weight $w$, such that $\{x,y\}$ and $\{u,v\}$ are in the same cycle, then $\{u,v\}$ is part of a minimum spanning tree.
(20 points)

5. For the following graph, find the residual network and the maximum flow.
(20 points)
1. (20 points) There are *two* things you must prove to show that a problem is NP-Complete. What are they?

2. (20 points) Suppose you are given an algorithm $A_k$ which solves the bounded bin-packing problem. The algorithm works as follows. If the number of items in the input is greater than $k$, the algorithm prints “Sorry, too many items.” Otherwise it tries every combination of the inputs to compute the optimal solution. Assuming that $k$ is fixed, prove that $A_k$ is a polynomially bounded algorithm. (i.e., $A_k$ is NOT NP-Complete, $A_k$ is in P.)

3. (20 points) Give a CREW-PRAM algorithm for computing the product of $n$ numbers.

4. (20 points) How many polynomials does FFT evaluate when computing a 16-point FFT? Give a general formula for an $n$ point FFT.

5. (20 points) The following is the KMP diagram for the string ABABABC. Show where the failure arcs go.

6. (20 points) NOTE: This problem was taken from the M.S. comprehensive examination given this spring.

   2-SAT is the set of boolean equations $e$ in conjunctive normal form such that each clause of $e$ has *exactly* 2 literals. Your friend from High-School has just completed the following proof that 2-SAT is NP-complete. (Note: $e$ must be of the form $(A \text{ OR } B) \text{ AND } (C \text{ OR } D) \text{ AND } ... \text{ AND } (E \text{ OR } G)$ where the letters A-G represent either a variable or its negation. $(A \text{ OR } B)$ is known as a clause of $e$.

   a. 2-SAT is obviously in NP, because SAT is in NP, and 2-SAT is a subset of SAT.

   b. The following transformation will convert 2-SAT to 3-SAT thus completing the proof. If $(A \text{ OR } B)$ is a clause of an expression $e$ replace $(A \text{ OR } B)$ with $(A \text{ OR } B \text{ OR } x_1) \text{ AND } (A \text{ OR } B \text{ OR } x_1')$. (Note $x_1'$ is the negation of $x_1$.) A new variable $x_i$ is created for each clause of $e$. The resulting equation $e'$ is satisfiable if and only if $e$ is satisfiable, and since 3-SAT is NP-Complete, so is 2-SAT.

   Now, it is well known that 2-SAT is in P, and is NOT NP-Complete. Therefore, your friend must have made a mistake. What was it?
1. Using these formulas:
\[(n+1)^2 = n^2 + 2n + 1\]
\[(n+1)^3 = n^3 + 3n^2 + 3n + 1\]
\[(n+1)^4 = n^4 + 4n^3 + 6n^2 + 4n + 1\]
Prove the following summation formula by induction.
(15 points).
\[\sum_{i=0}^{n} i^3 = \frac{n^4 + 2n^3 + n^2}{4}\]

2. Prove that for all real numbers \(a>0\) and \(b>0\) that if \(a>b\) then \(n^a\) grows faster than \(n^b\).
(15 points)

3. Prove that for any polynomial \(p(n) = p_i n^i + p_{i-1} n^{i-1} + ... + p_1 n + p_0\)
grows faster than \(\log_b n\). You may assume that if \(i>j\) that \(n^i\) grows faster than \(n^j\).
(15 points)

4. How many times does the function ABC print “HELLO WORLD” for an argument \(n \geq 0\)? Find a closed form function. (Hint, first find the recurrence relation, and solve it exactly.)
(20 points)

\[
\text{ABC}(n)
\begin{align*}
\text{if } (n > 0) & \text{ then} \\
& \text{PRINT “HELLO WORLD”;} \\
& \text{ABC}(n-1); \\
& \text{PRINT “HELLO WORLD”;} \\
& \text{ABC}(n-1); \\
& \text{endif}
\end{align*}
\]

5. Find the order of \(T(n)\) for the following recurrence relation, using the master method.
(20 points)

\[T(n) = 2T(n/4) + n^{1/2}\]
6. The following algorithm is the repeated minimum sort.

```
for i := 1 to n-1 do
    for j := i+1 to n do
        if L[i] > L[j] then
            temp := L[i];
            L[i] := L[j];
            L[j] := temp;
        endif
    endfor;
endfor
```

How many comparisons does this algorithm do in the Worst Case? In the Average Case? (Hint, think about the outer loop in reverse order, and then count how many times the inner loop executes for each iteration of the outer loop.)

(15 points)
1. In HEAPSORT, we used FIXHEAP to create the heap. FIXHEAP assumes that the heap condition holds at every vertex except, possibly, the root. An alternative procedure is to use the standard INSERT mechanism for creating the heap. The INSERT mechanism starts with a heap of size n and adds one element at the rightmost side of of the last level. (This is equivalent to increasing the size of the array by 1.) Then the element is moved upward into the heap until the heap condition is satisfied. The procedure is listed below. The ADJUST algorithm can be used in the heapify process as follows. Does this new HEAPIFY work? If so, is it more efficient or less efficient than the old HEAPIFY? Justify your answer.

(20 points.)

```plaintext
procedure HEAPIFY(ListSize:Integer);
var i:Integer;
beg
  for i:= 1 to ListSize do
    ADJUST(i);
endfor
end;
```

```plaintext
procedure INSERT(value, var heapsize:Integer)
begin
  heapsize := heapsize + 1;
  L[heapsize] := value;
  ADJUST(heapsize);
end;
```

```plaintext
procedure ADJUST(heapsize)
var ThisElement, temp: INTEGER;
beg
  ThisElement := heapsize;
  while (ThisElement > 1) and (L[ThisElement] > L[ThisElement div 2]) do
    temp := L[ThisElement];
    L[ThisElement] := L[ThisElement div 2];
    L[ThisElement div 2] := temp;
    ThisElement := ThisElement div 2;
endwhile
end;
```

2. Suppose that we have invented some miraculous new method of doing the SPLIT subroutine of QUICKSORT. This new SPLIT routine always divides the list so that 1/3 of the elements are in one half of the list and 2/3 are in the other half. What is the order of this new QUICKSORT algorithm? If you cannot give a precise proof of your answer, at least give a reasonable argument to support your claim.

(20 points)

3. Give an example to show that Dijkstra’s shortest path algorithm does not work if some edges have negative weights.

(20 points)

4. (a) We know that if no two edges of a graph have the same weight, then there is a unique minimum spanning tree. Give an example of a graph that has a unique minimum spanning tree, but has two or more edges of the same weight.
(b) Let $G$ be a graph with a cycle $(v_1, v_2, \ldots, v_n)$. The connecting edges would be $(v_1, v_2)$, $(v_2, v_3)$, ..., $(v_n, v_1)$. Assume that the heaviest edge in the cycle has weight $w$, and that two or more edges in the cycle have weight $w$. Draw an example of such a cycle. Prove that any graph consisting of such a cycle has two (or more) different minimum spanning trees.

(20 points)

5. When traversing the adjacency list of vertex $v$ in the Bi-Connected Component Depth First Search routine, we examined three possible conditions for the vertex $w$, adjacent to $v$. First, $w$ may be an ancestor of $v$ in the depth first search tree (including $v$'s parent). Second, $w$ may be a descendant of $v$ in the depth first search tree (not a child). Third, $w$ may be an unvisited vertex. What about the fourth possibility, namely that $w$ is in the depth first search tree, but is neither an ancestor nor a descendant of $v$? Why hasn’t any provision been made for this possibility?

(20 points)

6. On a separate sheet of paper, please comment on the matter discussed in class last time.
1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they?

2. (20 points) It is well known that the LONGEST PATH problem is NP-Complete. The LONGEST PATH problem is: Given a graph G, and two vertices A and B, what is the longest simple path from A to B? Professor Gooms from the University of Florida has come up with the following proof that P is equal to NP. “Take all the edge weights of G and negate them giving you a new weighted graph G’. Then use Dijkstra’s shortest path algorithm on G’ to find the shortest path. The shortest path in G’ is obviously the longest path in G, and Dijkstra’s algorithm runs in polynomial time, so P must be equal to NP.” Professor Gooms has screwed up (again). What did he do wrong?

3. (20 points) Give a CREW-PRAM algorithm for computing the inner product of two vectors. Recall that if V and W are two vectors such that V=(v_1,v_2, ..., v_n) and W=(w_1,w_2, ..., w_n) then the inner product of V and W, V•W, is (v_1*w_1)+(v_2*w_2)+ ... +(v_n*w_n).

4. (20 points) Find all the augmenting paths in the following graph. What is the maximum flow from s to t? Note: the capacity/flow numbers for a particular arc are located as close as possible to the beginning of the arc. If you are confused as to which numbers go with a particular arc, look at the beginning of the arc.
5. (20 points) Assuming that \( n \) is a power of two, how many times (as a function of \( n \)) will the following program print “Hellow Workld” for a given argument \( n \).

```pseudocode
Funny_Print(n:Integer)
begin
  if n \geq 1 then
    for i := 1 to c * n do
      Print("Hellow Workld");
    endfor
    Funny_Print(n/2);
    Funny_Print(n/2);
  endif
end
```

6. (20 points) As we all know, Radix sort is order \( n \), Quicksort is order \( n \times \lg n \), and Insertion sort is of order \( n^2 \). Nevertheless, for all practical values of \( n \), both Quicksort and Insertion sort have been found to be significantly faster than Radix sort. Why?
1. Using these formulas:
   \[(n+1)^2 = n^2 + 2n + 1\]
   \[(n+1)^3 = n^3 + 3n^2 + 3n + 1\]
   Prove the following summation formula by induction.
   (15 points).
   \[\sum_{i=1}^{n} (3i^2 + 4i + 2) = \frac{2n^3 + 5n^2 + 5n}{2}\]

2. Prove that the function \(f(n) = n \lg n\) grows faster than \(g(n) = n\), and slower than \(h(n) = n^\alpha\), where \(\alpha\) is some real number greater than 1.
   (15 points)

3. The basic operation for the function \(ABC\) is the number of times “HELLO WORLD” is printed? What is the time bound of \(ABC\)? Give both the recurrence relation that describes the behavior of \(ABC\), and then give the order.
   (20 points)
   \[
   \text{ABC}(n) \quad \text{if } (n > 1) \text{ then}
   \quad \text{for } i := 1 \text{ to } n \text{ do}
   \quad \text{for } j := i \text{ to } n \text{ do}
   \quad \text{PRINT “HELLO WORLD”;}
   \quad \text{PRINT “HELLO WORLD”;}
   \quad \text{endfor}
   \text{endfor}
   \text{ABC}(n/2);
   \text{ABC}(n/2);
   \text{ABC}(n/2);
   \text{endif}
   \]

4. Assuming that \(n=2^k\), for some integer \(k\), find the exact solution of the following recurrence relation.
   (20 points)
   \[T(n) = T(n/2) + n \lg n\]

5. Suppose we have written MergeSort in the conventional way, but have used a “canned” MERGE algorithm to do the merge operation. Much to our surprise we find out later that this MERGE algorithm does \(n^2\) comparison operations when the total length of the two lists is \(n\). What is the time bound of our algorithm?
   (15 points)

6. Suppose that you have discovered some miraculous new way to do quicksort that has the following properties. When you find the pivot point (or SPLITPOINT), it is always exactly in the center (or as close as you can get when the number of elements is even). Furthermore, the SPLIT operation, which still takes \(n-1\) comparisons, leaves the first half of the list sorted, while leaving the pivot point in its correct position and the last half of the list in random order. What is the time bound of this algorithm?
   (15 points)
1. The HEAPSORT algorithm studied in class is based on binary trees. The $n \log n$ time bound of the algorithm is based on the depth of the binary tree, which is $\log n$. How does the time bound of the algorithm change when binary trees are replaced with trees that have 3, 4 or more children per node? How does the number of comparisons-per-node change when running FIXHEAP? Does this affect the time bound? (20 points)

2. Devise an adversary for the problem of searching an ordered list. Your adversary may respond with three answers (EQUAL, LESS, or GREATER) when supplied with an index and a search-value. Use this adversary to show that binary search is optimal. Explicit code for the adversary is NOT required, but you must describe it in sufficient detail that I can believe it will work. (20 points)

3. Will the shortest path algorithm work if several of the edge-weights are zero? Why or why not? (20 points)

4. Professor Gooms from the University of Florida has told his algorithms class about his wonderful new algorithm for finding minimum spanning trees. “You just sort the edges by weight and choose the $n-1$ edges of minimum weight,” he claims. Professor Gooms has blown it (again). Give an example of a graph for which Professor Gooms’ algorithm won’t work. In your example, make sure that no two edges have the same weight. (20 points)

5. Suppose $G$ is a connected graph with at least three vertices. Suppose $x$ is a vertex of degree 1 which is connected to vertex $y$ by the edge $xy$. Show that vertex $y$ is an articulation point. (20 points)
1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain, in detail, how to do these two things.

2. (20 points) Transform the following CNF equation into 3-SAT form.

\[(A' \lor B) \land C' \land (A \lor B' \lor C' \lor D) \land (C \lor D \lor E')\]

3. (20 points) Give a CREW-PRAM algorithm for finding the MAX and MIN of a set of N numbers. How fast does your algorithm run?

5. (20 points) Using the algorithm in the book, find the strongly connected components of the following graph. For each vertex, show the value of the array “low,” and for each edge, indicate whether the edge is a tree edge, a back edge, or a cross edge. When the algorithm has a choice about which vertex to visit next, assume that the algorithm always chooses the lowest-numbered unvisited vertex. The search should start with vertex 1.
4. (20 points) The basic operation in the following algorithm is the “Print” statement. What is the order of this algorithm?

    Serious_Print(n:Integer)
    begin
        if n ≥ 1 then
            for i := 1 to n do
                Print("I am now doing a basic operation.");
            endfor
            for i := 1 to 3 do
                begin
                    for j := 1 to 3 do
                        Serious_Print(n/3);
                    endfor;
            endfor;
        endif
    end

6. (20 points) Convert the following matching problem to a maximum flow problem. Find one augmenting path, and show that this path represents an alternating path in the original problem. The heavy lines indicate matched vertices and edges.
1. Using these formulas:

\[(n+1)^2 = n^2 + 2n + 1\]
\[(n+1)^3 = n^3 + 3n^2 + 3n + 1\]

Prove the following summation formula by induction.
(15 points).

\[\sum_{i=1}^{n} (i^2 + 2i + 1) = \frac{2n^3 + 9n^2 + 13n}{6}\]

2. Prove that the function \( f(n) = n^k \log n \) grows faster than \( g(n) = n^k \), and slower than \( h(n) = n^{k+c} \), where \( c \) is any real number greater than 0.
(15 points)

3. The basic operation for the function ABC is the number of times “HELLO WORLD” is printed? What is the time bound of ABC? Give both the recurrence relation that describes the behavior of ABC, and then give the order.
(20 points)

```plaintext
ABC(n)
if (n > 1) then
  for i := 1 to n do
    j := n;
    while j > 1 do
      PRINT “HELLO WORLD”;
      PRINT “HELLO WORLD”;
      j := j / 2;
  endfor
  for i:= 1 to 8 do
    ABC(n/2);
  endfor
endif
```

4. Assuming that \( n=2^k \), for some integer \( k \), and that \( T(1) = 0 \), find the exact solution of the following recurrence relation.
(20 points)

\[T(n) = 4T(n/2) + n^2\]

5. Suppose you have invented a new version of Quicksort, and you find, much to your surprise, that your new version always selects the second largest element of the list as the pivot point. Prove that both the worst and average case of your new algorithm is \( \Theta(n^2) \).
(15 points)

6. Given a list of 10 or more elements, the problem is to find the second, fifth and seventh largest elements. Give both upper and lower asymptotic bounds on the complexity of this problem. The two bounds need not be equal, but you must prove that your bounds are correct.
(15 points)
1. Consider the following variation of Depth First Search.

DFS(V:Vertex)
   If V = TargetVertex Then
       Process Parent Array
   Else
       Mark and visit V;
       For every unmarked vertex W adjacent to V do
           Parent[W] = V;
           DFS(W);
       End For;
       Unmark V;
   End If
End DFS

TargetVertex = VertexA;
Parent[VertexB] = 0;
DFS(VertexB);

Show (not too detailed please) that for any simple path P tween VertexA and VertexB, that P will be processed by the statement “Process Parent Array.” Assume that the size of the input is equal to the number of vertices of the graph to be searched. Show that this algorithm is NOT polynomially bounded. (Hint consider a complete graph on n vertices.)

(20 points)

2. Consider the following bipartite matching problem. Show how to convert this to a maximum flow problem. Show a maximum matching. Show an augmenting path containing at least 5 vertices. The fat lines are the existing matches.

(20 points)
3. Suppose S and T are two minimum spanning trees for a graph G, and that S and T are identical except for one edge. In particular S has the edge \((u,v)\) which is missing from T, and T has the edge \((w,x)\), which is missing from S. Show that \((u,v)\) and \((w,x)\) must have the same weight.

(20 points)

4. Professor Gooms from the University of Florida has told his algorithms class about his wonderful new shortest path algorithm that will work for arbitrary graphs with negative edge weights. “First you find the negative edge weight with the largest absolute value. Call that absolute value \(k\). Then add \(k+1\) to the weight of each edge. Then find the shortest path using the usual algorithm.” Professor Gooms has blown it (again). Give an example of a graph for which Professor Gooms’ algorithm won’t work.

(20 points)

5. To show a problem is NP-Complete, you must prove two things. What are they? What does NP-Hard mean?

(20 points)
Complete All Problems on this page.

1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain, in detail, how to do these two things.

2. (20 points) A CNF equation contains the following clauses. What clauses would the equivalent 3-SAT form have?

\{X,Y,Z',W,V\}, \{Q,X,Y'W'\}, \{R,Q',T\}, \{Q,X'\} \{Z\}

3. (20 Points) Given the following 3-Sat clauses, show the truth setting component (variables A and B only) and the satisfaction testing component of the equivalent 3-dimensional matching problem.

\{A,B,C'\}, \{C',D',A'\}, \{B',D,C\}

4. (20 points) Give an optimal (in terms of time bound) CREW-PRAM algorithm for finding dot product of two vectors. (Component-wise multiplication, add up products.) How fast does your algorithm run? Could this be used to create a matrix multiplication algorithm? If so, how fast would it run?

5. (20 points) Assuming that \(N\) is a power of some suitable radix \(a\), how many times will the line “Hello World” be printed by the procedure Proc1 for the generic argument \(N\)? What is \(a\)? Note: Give only the order, not the exact solution!

```plaintext
procedure Proc1(N:Integer)
begin
  if N <= 1 then
    Print “Hello World”
  else
    for i := 1 to N step 2
      Print “Hello World”
    endfor
    for l := 1 to 4 do
      Proc1(N/9)
    endfor
  endif
end;
```
6. (20 points) You are working for a bicycle shop, and your employer has recently purchased a large number of chains and sprockets that he wants assembled into bicycles. Unfortunately, the chains and sprockets are all of odd gauges, so they are not interchangeable. After a great deal of effort, you have numbered all the chains and all the sprockets (there are an equal number of each) and you have determined which chains will work with which sprockets. Suggest a method for building the maximum number of bicycles from these parts. Is this problem NP-Complete? If so, suggest a suitable approximation.

7. (20 points) Suggest a suitable method for solving the following problem. If it is NP-Complete, suggest a suitable approximation method. In the great Central Iowa ballooning contest the objective is for the winning balloonist to carry as much weight as possible across the finish line. Each balloonist is given a collection of iron blocks of many different sizes. Each balloonist must load as many blocks as possible into the balloon and float across the finish line. Although it would be easy to fit all the blocks into the balloonist’s basket, the weight of all the blocks is much too large for any balloon. The problem is to find the maximum weight for each balloon. You may assume that the lifting capacity of each balloon is known, and that an adjustment has already been made for the pilot’s weight.

8. (20 points) Professor Gooms of the University of Florida claims to have discovered a new matrix multiplication method that requires only 3 multiplications and 4 additions to multiply a 2x2 matrix. There is something suspicious about this result. What is it? (Hint: Is there a known lower bound for Matrix Multiply?)

9. (20 points) Give the worst case time bound for each of the following sort algorithms.

   QuickSort, MergeSort, InsertionSort, HeapSort.

10. (20 points) Give an example that proves Dijkstra’s shortest path algorithm won’t work for graphs with negative edge-weights.
1. Prove the following by induction: A complete binary tree of height \( n \) has \( 2^{n+1} - 1 \) vertices.

2. Prove that the function \( f(n) = 2^n \) grows faster than \( g(n) = n^k \), for any real number \( k > 0 \).

(15 points)

3. The basic operation for the function ABC is printing “HELLO WORLD”. What is the asymptotic time bound of ABC? For \( n = 16777216 = 2^{24} \), how many times is “HELLO WORLD” printed? Give a formula that works for any power of 2.

(20 points)

```plaintext
ABC(n)
  if (n > 1) then
    for i := 0 to n do
      PRINT “HELLO WORLD”;
    endfor
    ABC(n/2);
  endif
end ABC
```

4. What is the order of \( T(n) \)? Why?

(20 points)

\[ T(n) = 4T(n/5) + n\sqrt{n} \]

5. Suppose you have invented a new version of Quicksort, which divides the list into small, medium, and large as illustrated in the following diagram. The list is always split into 3 equal pieces, but it takes \( n + \log n \) time to do so. What is the order of this new algorithm?

(15 points)

```
Small      Medium      Large
```

6. Mergesort uses the formula \( \text{Mid} = \lfloor (\text{First} + \text{Last})/2 \rfloor \) to split the list in two. Suppose you replace this formula with \( \text{Mid} = \lfloor (\text{First} + \text{Last})/4 \rfloor \). How does this affect the order of the Mergesort algorithm?

(15 points)
1. Let $G=(V,E)$ be a weighted graph, and let $G'=(V',E')$ be a subgraph of $G$, which is obtained by deleting vertices and their adjacent edges from $G$. In other words, if $(u,v) \in E$, and $u \in V', v \in V'$, then $(u,v) \in E'$. Let $T'$ be a minimum spanning tree of $G'$. Prove or disprove the following statement. $T'$ must be a subtree of some minimum spanning tree of $G$. (20 points.)

2. Give an example of a 6-vertex graph for which depth first search processes the vertices in the same order as breadth first search. (20 points.)

3. Consider the following bipartite matching problem. The fat lines are the existing matches. Show all alternating paths. Show a maximum match. (20 points)

4. Transform the following CNF-SAT equation into a 3-SAT equation, using the technique of the 3-SAT proof. (20 points.)

\[ (\overline{a} \lor b \lor c) \land \overline{e} \land (c \lor d) \land (c \lor \overline{d}) \lor \overline{x} \lor \overline{y} \lor \overline{z} \]

5. Complete the NP-Completeness proof of the 3DM (3 Dimensional Matching) problem by giving a nondeterministic polynomial time algorithm that solves the problem. (20 points)
1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain, in detail, how to do these two things.

2. (20 points) Assume you have an unlimited number of processors. Give an optimal CREW PRAM algorithm for computing the product of two square matrices. What is the time bound of your algorithm?

3. (20 points) Suppose you found a method for multiplying two matrices that took only five multiplications and twelve additions. How fast could you multiply two square matrices assuming the size was a power of two?

4. (20 Points) Prove that if Dijkstra’s shortest path algorithm would work for graphs with negative edge weights, then P=NP. (Hint, start with longest path, and think about altering the edge weights. How fast does Dijkstra’s algorithm run?)

5. (20 points) Assuming that \( N \) is a power of some suitable radix \( a \), how many times will the line “Hello World” be printed by the procedure Proc1 for the generic argument \( N \)? What is \( a \)?

Note: Give only the order, not the exact solution!

```pascal
procedure Proc1(N:Integer)
begin
  if N >= 1 then
    Print “Hello World”
    for i := 1 to N do
      for j := 1 to i do
        Print “Hello World”
      endfor
    endfor
    for i := 1 to 25 do
      Proc1(N/5)
    endfor
  endif
end;
```

6. (20 points) You are the elevator operator for the world’s tallest skyscraper the “Tampa Tower” which was recently built on the site of the old Bucs stadium. Unfortunately, the builders had to scrimp on the quality of the elevators due to a shortfall in funds. The elevators are huge enough to hold a hundred people, but can only lift 1000 pounds. One ounce more, and they get stuck for hours in some inaccessible place between the 12th and the 14th floors. Thousands of people come to visit this place every day. Your job is to get the people to the top in as few trips as possible. You are permitted to weigh each person, and you can assume that the fat guy from the circus won’t show up. Suggest an efficient algorithm for solving this problem. Point out any special difficulties that make this problem hard to solve. Your
algorithm should be optimal if possible, and a good approximation if an optimal solution isn’t feasible.
1. By induction, prove that for any $a$, $\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}$.

2. Using the definition of big-O notation, prove the following statement.
   Let $f(n) = 2^{p(n)}$ and $g(n) = 2^{q(n)}$, where $p(n)$ and $q(n)$ are functions such that $q(n) \in \omega(p(n))$. Then $g(n) \in \omega(f(n))$.
   (15 points)

3. Let $k>0$ be an integer, and $x>0$ be a real number. Show that $\log^k n$ grows slower than $n^x$.
   (20 Points)

4. The basic operation for the function ABC is printing “HELLO WORLD”. What is the asymptotic time bound of ABC? Give an exact formula that works for any power of 3.
   (20 points)

   ```
   ABC(n)
   if (n > 1) then
      PRINT “HELLO WORLD”;
      ABC(n/3);
      ABC(n/3);
      ABC(n/3);
   endif
   end ABC
   ```

5. You have invented a new sort algorithm, which breaks the list into three pieces using $\Theta(n)$ time. The Medium portion of the list is sorted, but the Small and Large portions are in random order. The Smalls are less than any Medium, and the Large’s are larger than any Medium. What is the order of this algorithm?
   (15 points)

6. Given a graph with $n$ vertices, what is the worst-case running time of depth-first search? Is breadth-first search any different? Why or why not?
   (15 points)
1. Let $G=(V,E)$ be a weighted graph. Let $T_1$ and $T_2$ be two minimum spanning trees of $G$, such that $T_1 \neq T_2$. Let $S_1$ be the set of edges that belong to $T_1$ and not to $T_2$. Let $S_2$ be the set of edges that belong to $T_2$ and not to $T_1$. Let $W_1$ be the total weight of all edges in $S_1$, and $W_2$ be the total weight of all edges in $S_2$. Prove that $S_1$ and $S_2$ have the same number of edges. Prove that $W_1 = W_2$. (20 points.)

2. Let $G=(V,E)$ be a rectangular graph as illustrated below, with all edge weights equal to 1. Suppose the shortest path between vertex $A$ and $B$ is of length $k$. What is the maximum and minimum number of vertices that must be visited before the Dijkstra algorithm finds the shortest path? (20 points.)

3. Show a graph, with starting and ending vertices $A$ and $B$. For this graph give an edge-processing order that will force the Bellman-Ford algorithm to complete the full $n-1$ iterations before computing the minimum path between $A$ and $B$. (20 points)

4. Transform the following VC problem into a Hamiltonian Path Problem. (20 points.)

5. Prove that the following problem is NP-Complete.  

**4-DIMENSIONAL MATCHING**

**INSTANCE:** Four Sets of Equal Size, $A$, $B$, $C$, and $D$, and a set of ordered 4-tuples $M \subseteq A \times B \times C \times D$.  

**QUESTION:** Is there a subset $C \subseteq M$, so that each element of $A$, $B$, $C$, and $D$ appears in exactly one element of $C$? (20 points)
1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain, in detail, how to do these two things.

2. (20 points) Assume you have an unlimited number of processors. Give an optimal CREW PRAM algorithm for computing an N-Point FFT.

3. (20 points) Suppose you found a method for multiplying two matrices that took only \( m \) multiplications and \( a \) additions. How fast could you multiply two square matrices assuming the size was a power of two?

4. (20 Points) Prove the following problem is NP-Complete. Given a graph \( G=(V,E) \), and an integer \( K \), is it possible to partition \( V \) into \( K \) disjoint sets \( V_1, \ldots, V_K \), such that each of the induced subgraphs is 3-Colorable?

5. (20 points) The basic operation for this algorithm is the Print “Hello World” statement. What is the order of the following algorithm? For an argument \( N \), what is the exact number of times

```pascal
procedure Proc1(N:Integer)
begin
  if N > 0 then
    Proc(N-1);
    Print "Hello World"
    Proc(N-1);
  endif
end;
```

6. (20 points) Transform the following 3-Sat formula into vertex cover problem.

\[(A+B'+C)(C'+D+E')(D'+B+E)\]
1. Find a closed form function for the following summation. \( f(n) = \sum_{i=1}^{n} (2i^2 + 3i + 7) \).

(15 Points.)

2. Using the definition of big-O notation, prove that the function \( f(n) = 2^n \) grows faster than \( g(n) = n^n \).

(15 points)

3. Let \( k > 0 \) be an integer, and \( x > 0 \) be a real number. Show that \( \log^k n \) grows slower than \( n^x \).

(20 Points)

4. The basic operation for the function ABC is printing “HELLO WORLD”. Give a recurrence relation that describes the performance of ABC. What is the asymptotic time bound of ABC?

(20 points)

```plaintext
ABC(n)
    if (n > 1) then
        for I = 1 to n do
            for j = 1 to I do
                for k = 1 to n do
                    PRINT “HELLO WORLD”;
                Endfor
            Endfor
        Endfor
    for I = 1 to 5 do
        ABC(n/7);
    Endfor
end ABC
```

5. You have invented a new merge algorithm that can merge two lists in place using only one comparison. If you implement MergeSort using this algorithm, how fast will it be?

(15 points)

6. What is the asymptotic running time of Prim Minimum Spanning Tree algorithm on a complete graph with \( n \) vertices. Answer the same question for Kruskal’s algorithm.

(15 points)
1. Let $G=(V,E)$ be a weighted graph. Suppose $G$ has a cycle $v_1, v_2, \ldots, v_k$. Suppose that in this cycle there are two different edges with the same weight, $(v_i,v_{i+1})$ and $(v_j,v_{j+1})$. Suppose that a minimum spanning tree has been found that includes all edges of the cycle except $(v_i,v_{i+1})$. Prove that $G$ has at least two minimum spanning trees. (20 points.)

2. Suppose you were to start Dijkstra’s shortest path algorithm on vertex 1 of a graph, and let it run until all vertices were added to the tree. Would this produce a minimum spanning tree? Prove your answer. Be rigorous in your proof. (20 points.)

3. Professor Gooms has made the following three statements. These statements are either false or utter nonsense. For each, explain why.
   a. “When Professor Licht says that the BZRQ problem is NP-Complete, what he really means is that his algorithm for the BZRQ problem is NP-Complete.”
   b. “The P≠NP problem is trivial. Of course P≠NP. To determine whether a Boolean expression is satisfiable, you have to search through an exponential number of items, and how could you possibly do that in polynomial time?”
   c. “I have shown that P≠NP! I can transform the satisfiability problem into the GOOMBAH problem in polynomial time, and last week Professor Drupe proved that the GOOMBAH problem requires an exponential amount of time.” (Curiously enough, it turns out that professor Drupe was correct, and that Professor Gooms’ transformation is also correct.)
   (20 points)

4. Transform the following Boolean expression into a vertex cover problem. (20 points.)
   $$(a' \lor b' \lor c) \land (a' \lor b \lor d') \land (a \lor c' \lor e) \land (b' \lor c' \lor e) \land (a \lor d' \lor e')$$

5. Prove that the following problem is NP-Complete.
   **N-DIMENSIONAL PARTITION**
   **INSTANCE:** A set of objects $A=\{a_1,a_2,\ldots,a_k\}$, a sizing function $s(a_i)$ which gives the size of each object, and an integer $K$.
   **QUESTION:** Is there a way to partition $A$ into $K$ distinct subsets $B_1 \subseteq A$, $B_2 \subseteq A \ldots B_K \subseteq A$, such that for $i \neq j$, $B_i \cap B_j = \emptyset$, and the sum of the sizes of the objects in each of the sets is the same? (20 points)
1. (20 points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain, in detail, how to do these two things.

2. (20 points) Given an square matrix $M$ and a real value $X$ we want to compute a value for each element of the square matrix. After computation, element $M[i,j]$ should contain the value $X_{ij}$. Using a CREW PRAM, how fast can you compute all values of $M$? Is the answer different for an EREW PRAM?

3. (20 points) Suppose you have discovered a method of multiplying a 2x2 matrix that requires only 3 multiplications and 10 additions. Using Strassen’s method, what would the time bound of the new algorithm be?

4. (20 Points) Prove the following problem is NP-Complete. Given a set of objects $\{x_1, x_2, \ldots, x_n\}$, and a real number $K > 0$, and a size function $s$ which is defined for each object $x_i$, with $s(x_i) = r_i$ for some real number $K \geq r_i > 0$, attempt to fit the given objects into the fewest possible bins of size $K$.

5. (20 points) Do this problem last. The basic operation for this algorithm is the Print “Hello World” statement. What is the order of Proc1? Give the precise function, $F(N)$, that gives the number of times “Hello World” is printed for each value of $N$, including those values that are not exact powers of 2. If a closed form expression cannot be obtained, give an English description of the function. Explain why $F(100) = 197$.

```plaintext
procedure Proc1(N:Integer)
begin
  if N > 0 then
    For I := 1 to N Do
      Print “Hello World”;
    EndFor
    Proc1(N/2); {Integer Division}
  Endif;
end;
```
6 (20 points) You are working for a shipping company that specializes in moving heavy objects. You are asked to create a program that a truck driver can use to pack his truck as efficiently as possible. The company’s trucks have many different carrying capacities, and a driver may drive different trucks on different days. The objective is to feed the carrying capacity of the vehicle (in pounds) into the program, then the weights of the various objects that need to be transported, and pack as much weight into the truck as possible. It will seldom be possible to pack everything into a single truck. If this problem is NP-Complete, explain why, and suggest an approximation method. If it is not NP-complete, suggest a method that will determine the exact solution.
1. Find a closed form function for the following summation. \( f(n) = \sum_{i=1}^{n} i^3 \).
(20 Points.)

2. Prove that the function \( f(n) = 3.7^n \) grows faster than \( g(n) = 2.9^n \).
(20 points)

3. The basic operation for the function ABC is printing “HELLO WORLD”. Give a recurrence relation that describes the performance of ABC. What is the asymptotic time bound of ABC?
(20 points)

```plaintext
ABC(n)
  if (n > 1) then
    for I = 1 to n do
      for j = 1 to I do
        for k = 1 to j do
          for m = 1 to k do
            PRINT “HELLO WORLD”;
          Endfor
        Endfor
      Endfor
    endfor
  endif
end ABC
```

1.5, 2.25, 3.375, 5.0625, 7.59375, 11.390625, 17.0859375, 25.62890625, 38.443359375, 57.6650390625, 86.49755859375

4. You have invented a new version of mergesort that divides the list into three pieces, calls itself recursively on the three pieces, and then uses a three-way merge to create the final list. Is this algorithm faster or slower than the original? Why?
(20 points)

5. In a directed graph, the in-degree of a vertex is the number of edges entering the vertex, while the out-degree is the number of edges leaving the vertex. In a directed acyclic graph prove that there is at least one vertex with in-degree zero, and at least one vertex with out-degree zero.
(20 points)
1. Let $G=(V,E)$ be a weighted (undirected) graph. Let $G_k=(V,E_k)$ be the graph that is obtained by adding the positive constant $k$ to the weight of every edge of $G$. Prove or disprove: Every minimum spanning tree of $G$ is a minimum spanning tree of $G_k$. (20 points.)

2. Let $G=(V,E)$ and $G_k=(V,E_k)$ be as defined above. Prove or disprove: For every pair of vertices $A, B \in V$, every shortest path in $G$ between $A$ and $B$, is also a shortest path in $G_k$. (20 points.)

3. There are two things you must prove to show that a problem is NP-Complete. What are they? (20 points)

4. Transform the following CNF into a 3-CNF expression using the SAT=>3-SAT transformation presented in class. (20 points.)

$$(a' \lor b' \lor c \lor d') \land (a' \lor b \lor d') \land (c' \lor e) \land (e) \land (a \lor d' \lor e' \lor c' \lor d' \lor q)$$

5. Convert this matching problem into a maximum flow problem, and find the maximum matching. Show your work. (20 points)
1. (20 Points) There are two things you must prove to show that a problem is NP-Complete. What are they? Explain these two things in detail.

2. (20 Points) Professor Gooms from the University of Florida has invented a new sort algorithm which is identical to Merge-Sort in every respect, except his Merge algorithm runs in $\Theta(n^2)$ time instead of $\Theta(n)$. He claims that this makes his algorithm have an $\Theta(n^2 \lg n)$ time bound instead of $\Theta(n \lg n)$. Is he correct? Why or why not. BE SPECIFIC!

3. (20 Points) **Proof that P=NP.** We know that we can solve the marriage problem (2D matching) in polynomial time. But 2D-Matching is obviously NP-Complete, for the following reason. Given a 2DM problem, we could non-deterministically guess at a solution, and then test for validity in polynomial time, so 2DM is in NP. It is also easy to transform any 2DM problem into a 3DM problem by taking each edge in the 2DM problem, (u,v), and transforming it into a triple (u,v,v'). The new problem has a complete match if and only if the original has a complete match. Comment on the correctness of this proof.

4. (20 Points) Given two one-dimensional arrays A and B, both of size n, write a CREW PRAM (or more restrictive) algorithm for computing


   Your algorithm should run in lg n time.

5. (20 Points) In a directed graph, any vertex with indegree 0 is called a source while any vertex with outdegree 0 is called a sink. Let G=(V,E) be a strongly connected graph with at least two vertices. Prove that G has no sources and no sinks.

6. (20 Points) An example of a vertex cover problem is “Does the following graph have a vertex cover of size 2?” Transform this vertex cover problem into a Hamiltonian Path problem using the transformation presented in class.
1. Prove that the following equation holds for all $n$. Use induction.

$$\sum_{i=0}^{n} 3^i = \frac{3^{n+1} - 1}{2}$$

(20 Points.)

2. Prove that the function $f(n) = \lg^2 n$ grows slower than $g(n) = n^{0.025}$.

(20 points)

3. Determine which function grows faster, $f(n) = 0.1^n$ or $g(n) = n^{10^{10}}$.

4. The basic operation for the function ABC is printing “HELLO WORLD”. Give a recurrence relation that describes the performance of ABC. What is the asymptotic time bound of ABC?

(20 points)

```
ABC(n)
    if (n > 1) then
        for I = 1 to n do
            for j = 1 to I do
                PRINT “HELLO WORLD”;
            Endfor
        endfor
        for I = 1 to 18 do
            ABC(n/3);
        endfor
    endif
end ABC
```

5. You have developed an on-line system that transmits transactions in bursts of 20. You want to accumulate these transactions into a sorted list (sorted into ascending order by time). The transactions in a burst are all mixed up, and must be sorted, but the bursts are transmitted in order. So all transactions in burst 1 are earlier than all transactions in burst 2, and so forth. You decide to use insertion sort to sort the transactions. How much time will it take to sort $n$ transactions? More precisely, what is the asymptotic time bound of this algorithm. (Warning: I expect something more clever than $\frac{n(n-1)}{2}$.)

(20 points)
1. Let G=(V,E) be a directed graph. What is the maximum number of high-level calls that must be made when performing a Depth First Search of G? Prove your answer. Suggest a method for finding the minimum number of calls.  
(10 points)

2. In the Strongly Connected Component Algorithm presented in class (Not in the book!) is it possible to use breadth first search instead of depth first search? Prove your answer.  
(10 points)

3. Let G be a graph with two adjacent vertices A and B. Removing the edge (A,B) causes the graph to become disconnected. Does this mean that A and B are articulation points? Prove your answer?  
(10 points)

4. Let G be a weighted graph. G is a simple cycle. Write an algorithm for calculating the number of Minimum Spanning Trees of G.  
(10 points)

5. Let G be a directed graph. Run the following algorithm, AlgorithmX, on G.
   AlgorithmX: Begin
   for i:= 1 to n do
     Mark[i] := False;
   endfor
   Mark[j] := True;
   for t := 1 to k do
     for i := 1 to n do
       Ptr := AdjList[i];
       while Ptr = NULL do
         w := Ptr->vertex;
         if Mark[i] then
           Mark[w] := True;
         endif
       endwhile
     endfor
   endfor
   End

Which vertices of G are guaranteed to be marked when AlgorithmX finishes. Prove your answer. For a specific graph G, is it possible to guarantee that every vertex of G will be marked? Prove your answer. What must k equal if there is to be a chance of marking every vertex of G?  
(10 points)
6. Let G be a weighted graph with vertices A, B, C, D, and E, as well as several others. The shortest path from A to E, in terms of edges, is the path A,B,C,D,E. Suppose B, C, and D are articulation points of G. Does this imply that the path A,B,C,D,E is the shortest path from A to E? Prove your answer.

(10 points)

7. When finding an augmenting path for a maximum flow problem, it is sometimes necessary to move backwards along an edge to find an augmenting path. However, it is never necessary to go all the way back to the source. Prove it.

(10 points)

8. Solve the following recurrence relations. Growth rate is sufficient.

\[ T(n) = 7 T\left(\frac{n}{10}\right) + n \lg n \quad \text{T(n)} = 8 T\left(\frac{n}{4}\right) \]
\[ T(n) = 27 T\left(\frac{n}{2}\right) + n^3 \quad \text{T(n)} = 625 T\left(\frac{n}{5}\right) + n^4 \]
\[ T(n) = 216 T\left(\frac{n}{6}\right) + n^4 \quad \text{T(n)} = 2 T\left(\frac{n}{2}\right) + n^2 \]
\[ T(n) = 2 T\left(\frac{n}{3}\right) + n^2 \quad \text{T(n)} = 2 T\left(\frac{n}{4}\right) + \sqrt{n^3} \]
\[ T(n) = 17 T\left(\frac{n}{3}\right) + n^2 \quad \text{T(n)} = 22 T\left(\frac{n}{4}\right) + n^2 \]

(10 points)

9. Differentiate the following functions.

\[ f(n) = n^{\lg 7} \quad f(n) = \sqrt[3]{e^{\lg n} + n^3} \]
\[ f(n) = \binom{n}{3} \quad f(n) = \lg^k n \]
\[ f(n) = \sqrt[3]{7^{\lg n}} \quad f(n) = \sum_{i=0}^{\infty} n^i \]

(10 points)

10. Order the following functions by growth rate.

\[ f(n) = n^{\lg 7} \quad f(n) = \sqrt[3]{e^{\lg n} + n^3} \]
\[ f(n) = \binom{n}{3} \quad f(n) = \lg^k n \]
\[ f(n) = \sqrt[3]{7^{\lg n}} \quad f(n) = \sum_{i=0}^{\infty} n^i \]

(10 points, all or nothing)
1. (10 Points) For the Partitioned Vertex Cover Problem (PVC), you are given a graph \( G=(V,E) \), and a sequence of integers \( C=(C_1,C_2, \ldots, C_k) \), \( k \geq 1 \). You must determine whether it is possible to partition \( V \) into \( k \) subsets, \( V_1, \ldots, V_k \) such that the subgraph induced by \( V_i \) has a vertex cover of size \( C_i \) or less. Prove that PVC is NP-Complete.

2. (10 Points) For the Row/Column Selection problem (RCS), you are given an \( n \)-by-\( n \) matrix \( M \) and a positive integer \( K \). Some entries of the matrix are zero and some are non-zero. The problem is to select a subset \( S=(S_1,\ldots,S_K) \) of the integers where \( 1 \leq S_i \leq n \), such that every non-zero element of \( M \) is either row \( S_j \) or column \( S_j \), for some \( S_j \in S \). Is RCS NP-Complete? If so prove it. If not, give a polynomial time algorithm for it.

3. (10 Points) Professor Gooms from the University of Florida has found a new approximation algorithm for the bin-packing problem. He claims that for his approximation, which runs in polynomial time, that the total amount of free space in all bins is guaranteed to be less than 0.85. There is something suspicious about this claim. What is it?

4. (10 Points) Let \( G=(V,E) \) be a directed graph. Prove or disprove the following statement. A directed graph is acyclic if and only if each vertex of \( G \) is a strongly connected component of \( G \).

5. (10 Points) Let \( G=(V,E,W) \) be a weighted graph, and let \( A, B \) and \( C \) be two vertices of \( G \). Let the shortest path from \( A \) to \( B \) be \( A,V_1,V_2,V_3, \ldots, V_n, B \), and \( B, W_1, W_2, \ldots, W_m, C \). Prove or disprove: \( A,V_1,V_2,V_3, \ldots, V_n, B, W_1, W_2, \ldots, W_m, C \) is the shortest path from \( A \) to \( C \).

6. (10 Points) Solve the following recurrence relations. GIVE ONLY THE TIME BOUND!
   a. \( T(n)=2T(n-1)+1 \)
   b. \( T(n)=16T(n/4)+n^2 \)
   c. \( T(n)=10T(n/7)+n \lg n \)
   d. \( T(n)=7T(n/10)+n \lg n \)
   e. \( T(n)=T(n-1)+n^4 \)
1. Prove that the following equation holds for all \( n \), and for all positive integers \( a \). Use induction.

\[
\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}
\]

(20 Points.)

2. Determine which grows faster. Justify your result. \( f(n) = n^{1.25} \) or \( g(n) = n^{1.22} \lg^{12} n \). (10 points)

3. Determine which function grows faster and prove your result. \( f(n) = e^{2\lg n} \) or \( g(n) = 2^{4n+1} \). (15 Points)

4. The basic operation for the function ABC is printing “HELLO WORLD”. Give a recurrence relation that describes the performance of ABC. What is the asymptotic time bound of ABC? (20 points)

```latex
\text{ABC}(n)
\quad \text{if } (n > 1) \text{ then}
\quad \quad k = 1
\quad \quad \text{while } (k<n) \text{ do}
\quad \quad \quad \text{PRINT “HELLO WORLD”;
\quad \quad \quad k = k * 3;
\quad \quad \text{endwhile}
\quad \text{for } i = 1 \text{ to } 10 \text{ do}
\quad \text{ABC}(n/5);
\quad \text{endfor}
\quad \text{endif}
\end{ABC}
```

5. You have a new sort algorithm that divides a list into three equal parts. Splitting the list requires \( \Theta(n) \) time. The center portion of the list is already sorted and in its correct position. The algorithm is called recursively on the first and last portions of the list. What is the asymptotic time bound of this algorithm? (20 points)

6. Find the asymptotic time bound of the following function. (15 points)

\[
f(n) = \sum_{i=1}^{n} e^i
\]