Recurrence Relations

Analyzing the performance of recursive algorithms
Worst Case Analysis

- Select a Hypothetical Set of Inputs
- Should be expandable to any size
- Make Algorithm run as long as possible
- Must not depend on Algorithm Behavior
  - May use static behavior analysis
  - Inputs may not change dynamically
Average Case Analysis

- Determine the range of inputs over which the average will be taken
- Select a probability distribution
- Characterize Amount of work required for each input
- Compute the Mean Value
**Binary Search**

BinS(X:Item;First,Last:Integer):Integer
var Middle:Integer;
begin
  if First <= Last then
    Middle = (First + Last)/2;
    if L[Middle] = X then return Middle;
    if L[Middle] < X then return BinS(X,Middle+1,Last);
    if L[Middle] > X then return BinS(X,First,Middle-1);
  else
    return -1;
  endif;
end;
Analysis of BinS: Basis

- If List Size = 1 then First = Last
- If List Size = 0 then First > Last
  if First <= Last then ...
  else return -1;
- Then $W(0) = 0$
Analysis of BinS: Recursion I

- If comparison fails, find size of new list
- If list size is odd:
  - First = 1, Last = 5, Middle = (1 + 5) / 2 = 3
- If list size is even:
  - First = 1, Last = 6, Middle = (1 + 6) / 2 = 3
Exercise: Verify that these examples are characteristic of all lists of odd or even size.

Worst case: Item X not in list, larger than any element in list.

If list is of size $n$, new list will be of size:\n\[\left\lfloor \frac{n}{2} \right\rfloor\]
Analysis of BinS: Recurrence

W(0) = 0

W(n) = 1 + W(\lfloor n / 2 \rfloor)

Is W(n) ∈ Θ(n)?
Solving the Recurrence

\[ W(n) = 1 + W(\lfloor n / 2 \rfloor) \]

We need to eliminate \( W \) from the RHS.

\[ W(\lfloor n / 2 \rfloor) = 1 + W(\lfloor \lfloor n / 2 \rfloor / 2 \rfloor) = 1 + W(\lfloor n / 4 \rfloor) \]

\[ W(n) = 1 + 1 + W(\lfloor n / 4 \rfloor) \]
Continuing the Exercise ...

\[ W(n) = 1 + W(\lfloor n / 2 \rfloor) \]
\[ W(n) = 1 + 1 + W(\lfloor n / 4 \rfloor) \]
\[ W(n) = 1 + 1 + 1 + W(\lfloor n / 8 \rfloor) \]
\[ W(n) = 1 + 1 + 1 + 1 + W(\lfloor n / 16 \rfloor) \]
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\lfloor n / 2^k \rfloor \right) \]
Ah Ha! A General Formula!

$$W(n) = k + W\left(\left\lfloor n / 2^k \right\rfloor\right)$$

But, if $k$ is big enough ...
Ah Ha! A General Formula!

\[ W(n) = k + W(\left\lfloor n / 2^k \right\rfloor) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...
Ah Ha! A General Formula!

\[ W(n) = k + W(\lfloor n / 2^k \rfloor) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...

Since \( W(0) = 0 \) ...
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\lfloor n / 2^k \rfloor\right) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...

Since \( W(0) = 0 \) ...

The \( W \) on the RHS ...

Ah Ha! A General Formula!

\[ W(n) = k + W(\lfloor n / 2^k \rfloor) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...

Since \( W(0) = 0 \) ...

The \( W \) on the RHS ...

Will Disappear!
When Will $W$ disappear?

When:

$$2^{k-1} \leq n < 2^k$$
When Will W Disappear?

When:
\[2^{k-1} \leq n < 2^k\]

Take the Log:
\[k - 1 \leq \lg n < k\]
When Will W Disappear?

When:
\[ 2^{k-1} \leq n < 2^k \]

Take the Log:
\[ k - 1 \leq \lg n < k \]

Take the Floor:
\[ k - 1 = \lfloor \lg n \rfloor \]
\[ k = \lfloor \lg n \rfloor + 1 \]
Finally ...

\[ W(n) = k + W(\lfloor n / 2^k \rfloor) \]

\[ W(n) = \lceil \lg n \rceil + 1 + W(0) \]

\[ W(n) = \lfloor \lg n \rfloor + 1 \]
Some Other Recurrences

\[ Q(n) = n - 1 + 2 Q\left(\frac{n}{2}\right) \]

\[ Q(1) = 0 \]

\[ W(n) = cn + W\left(\frac{n}{2}\right) \]

\[ W(1) = 1 \]
Recurrences with $T(1)=1$

\[
T(n) = T(n/2) + c \lg n
\]

\[
T(n) = T(n/2) + cn
\]

\[
T(n) = 2T(n/2) + cn
\]

\[
T(n) = 2T(n/2) + cn \lg n
\]

\[
T(n) = 2T(n/2) + cn^2
\]
The Challenger

\[ T(n) = \sqrt{n} \, T(\sqrt{n}) + cn \]

\[ T(2) = 1 \]
The Master Theorem I

\[ T(n) = a \cdot T(n / b) + f(n) \]

if \( f(n) \in O(n^{\log_b a - \varepsilon}) \), for some \( \varepsilon > 0 \)

\[ T(n) \in \Theta(n^{\log_b a}) \]
The Master Theorem II

\[ T(n) = aT(n/b) + f(n) \]

if \( f(n) \in \Theta(n^{\log_b a}) \)

\[ T(n) \in \Theta(n^{\log_b a \log n}) \]
The Master Theorem III

\[ T(n) = aT\left(\frac{n}{b}\right) + f(n) \]

if \( f(n) \in \Omega\left(n^{\log_b a + \varepsilon}\right) \), for some \( \varepsilon > 0 \)

and \( af(n/b) \leq cf(n) \) for some \( c < 1 \)

(for sufficiently large \( n \))

\[ T(n) \in \Theta(f(n)) \]
Homework

- Page 72, 4-1
- Page 73, 4-4
Recurrence Relations

Analyzing the performance of recursive algorithms
Worst Case Analysis

- Select a Hypothetical Set of Inputs
- Should be expandable to any size
- Make Algorithm run as long as possible
- Must not depend on Algorithm Behavior
  - May use static behavior analysis
  - Inputs may not change dynamically
**Average Case Analysis**

- Determine the range of inputs over which the average will be taken
- Select a probability distribution
- Characterize Amount of work required for each input
- Compute the Mean Value
Binary Search

BinS(X:Item;First,Last:Integer):Integer
var Middle:Integer;
begin
if First <= Last then
  Middle = (First + Last)/2;
  if L[Middle] = X then return Middle;
  if L[Middle] < X then return BinS(X,Middle+1,Last);
  if L[Middle] > X then return BinS(X,First,Middle-1);
else
  return -1;
endif;
end;
Analysis of BinS: Basis

- If List Size = 1 then First = Last
- If List Size = 0 then First > Last
  if First <= Last then ...
  else return -1;
- Then W(0) = 0
Analysis of BinS: Recursion I

- If comparison fails, find size of new list
- If list size is odd:
  - First=1, Last=5, Middle=(1+5)/2=3
- If list size is even:
  - First=1, Last=6, Middle=(1+6)/2=3
Analysis of BinS: Recurrssion II

- Exercise: Verify that these examples are characteristic of all lists of odd or even size.
- Worst case: Item X not in list, larger than any element in list.
- If list is of size n, new list will be of size: \[
\left\lfloor \frac{n}{2} \right\rfloor
\]
Analysis of BinS: Recurrence

\[ W(0) = 0 \]

\[ W(n) = 1 + W(\lfloor n / 2 \rfloor) \]

Is \( W(n) \in \Theta(n) \)?
Solving the Recurrence

\[ W(n) = 1 + W(\lfloor n / 2 \rfloor) \]

We need to eliminate \( W \) from the RHS.

\[ W(\lfloor n / 2 \rfloor) = 1 + W(\lfloor \lfloor n / 2 \rfloor / 2 \rfloor) \]

\[ = 1 + W(\lfloor n / 4 \rfloor) \]

\[ W(n) = 1 + 1 + W(\lfloor n / 4 \rfloor) \]
Continuing the Exercise ...

\[
W(n) = 1 + W(\lfloor n / 2 \rfloor)
\]

\[
W(n) = 1 + 1 + W(\lfloor n / 4 \rfloor)
\]

\[
W(n) = 1 + 1 + 1 + W(\lfloor n / 8 \rfloor)
\]

\[
W(n) = 1 + 1 + 1 + 1 + W(\lfloor n / 16 \rfloor)
\]
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\left\lfloor \frac{n}{2^k} \right\rfloor \right) \]
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\lfloor n / 2^k \rfloor \right) \]

But, if \( k \) is big enough ...
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\left\lfloor n / 2^k \right\rfloor\right) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...
Ah Ha! A General Formula!

\[ W(n) = k + W\left(\left\lfloor n / 2^k \right\rfloor\right) \]

But, if \( k \) is big enough ...

Then the argument of \( W \) will be zero, and ...

Since \( W(0) = 0 \) ...
Ah Ha! A General Formula!

$$W(n) = k + W\left(\lfloor n / 2^k \rfloor \right)$$

But, if $k$ is big enough ...

Then the argument of $W$ will be zero, and ...

Since $W(0) = 0$ ...

The $W$ on the RHS ...
Ah Ha! A General Formula!

\[ W(n) = k + W(\lfloor n / 2^k \rfloor) \]

But, if \( k \) is big enough ...
Then the argument of \( W \) will be zero, and ...
Since \( W(0) = 0 \) ...

The \( W \) on the RHS ...

Will Disappear!
When Will W disappear?

When:

$$2^{k-1} \leq n < 2^k$$
When Will W Disappear?

When:
\[ 2^{k-1} \leq n < 2^k \]

Take the Log:
\[ k - 1 \leq \lg n < k \]
When Will W Disappear?

When:
\[ 2^{k-1} \leq n < 2^k \]

Take the Log:
\[ k - 1 \leq \lg n < k \]

Take the Floor:
\[ k - 1 = \lfloor \lg n \rfloor \]
\[ k = \lfloor \lg n \rfloor + 1 \]
Finally ...

\[ W(n) = k + W\left(\lfloor n / 2^k \rfloor \right) \]

\[ W(n) = \lfloor \lg n \rfloor + 1 + W(0) \]

\[ W(n) = \lfloor \lg n \rfloor + 1 \]
Some Other Recurrences

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

\[ W(n) = cn + W(n/2) \]
\[ W(1) = 1 \]
Recurrences with $T(1)=1$

$T(n) = T(n / 2) + c \lg n$

$T(n) = T(n / 2) + cn$

$T(n) = 2T(n / 2) + cn$

$T(n) = 2T(n / 2) + cn \lg n$

$T(n) = 2T(n / 2) + cn^2$
The Challenger

\[ T(n) = \sqrt{n} T(\sqrt{n}) + cn \]

\[ T(2) = 1 \]
The Master Theorem I

\[ T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) \]

if \( f(n) \in O\left(n^{\log_b a - \varepsilon}\right) \), for some \( \varepsilon > 0 \)

\[ T(n) \in \Theta\left(n^{\log_b a}\right) \]
The Master Theorem II

\[ T(n) = a T\left(\frac{n}{b}\right) + f(n) \]

if \( f(n) \in \Theta(n^{\log_b a}) \)

\[ T(n) \in \Theta(n^{\log_b a \lg n}) \]
The Master Theorem III

\[ T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) \]

if \( f(n) \in \Omega\left(n^{\log_b a + \varepsilon}\right) \), for some \( \varepsilon > 0 \)

and \( af\left(\frac{n}{b}\right) \leq cf(n) \) for some \( c < 1 \)

(for sufficiently large \( n \))

\[ T(n) \in \Theta(f(n)) \]
Homework

- Page 72, 4-1
- Page 73, 4-4