Week 10 Reminders

Test 2 next week
practise test on webpage

HW4 (cast one!) out early next week

Induction
asymptotic analysis of functions $O, \Omega, \Theta$
Runtime analysis of algorithms $BC, WC$
Now we want to consider algorithms that have multiple inputs of length \( n \).

How to measure runtime?

\[ A(n) \]

\[ \text{Runtime} \]

\[ A=1 \quad A=2 \quad A=3 \]

\[ \text{WC}_A(n) \]

\[ \text{BC}_A(n) \]
Now we want to consider algorithms that have multiple inputs of length $n$. How to measure runtime?

$A(n)$

$WC_A(n)$

$BC_A(n)$

$BC_A(n) \in O(g)$

$BC_A(n) \in \Omega(g)$
More formally, to show $\text{WC}_f(n) \leq O(g(n))$:

we have to show

$\exists n_0, c_0 \forall n \geq n_0 \quad \text{WC}_f(n) \leq c_0 \cdot g(n)$

$= \exists n_0, c_0 \forall n \geq n_0 \quad \max \{ \text{Runtime}_f(x) \} \leq c_0 \cdot g(n)$

inputs $x$ of length $n$

$= \exists n_0, c_0 \forall n \geq n_0 \quad \forall x \quad \left[ \text{If } x \text{ has length } n \Rightarrow \text{Runtime}_f(x) \leq c_0 \cdot g(n) \right]$.

In general to prove $\text{WC}_f(n) \leq O(g(n))$ you usually do a static analysis of code.

For $\text{WC}_{\text{has-even}}(n) \leq O(n)$

since $n_0 = 1$, $c_0 = 1$. Argue for any list of length $n$, the for loop executes at most $n$ times so runtime on all $x$, $|x| = n$ is $n$
More formally, to show \( WC_f(n) \leq O(g(n)) \):

We have to show:

\[
\forall n_0, c_0 \quad \forall n \geq n_0 \quad WC_f(n) \leq c_0 \cdot g(n)
\]

\[
= \forall n_0, c_0 \quad \forall n \geq n_0 \quad \left[ \max_{x \mid |x| = n} \{ \text{Runtime}_f(x) \} \leq c_0 \cdot g(n) \right]
\]

Similarly to show \( WC_f(n) \leq \Omega(g(n)) \):

We have to show:

\[
\forall n_0, c_0 \quad \forall n \geq n_0 \quad WC_f(n) \geq c_0 \cdot g(n)
\]

\[
= \forall n_0, c_0 \quad \forall n \geq n_0 \quad \left[ \max_{x \mid |x| = n} \{ \text{Runtime}_f(x) \} \geq c_0 \cdot g(n) \right]
\]

\[
= \forall n_0, c_0 \quad \forall n \geq n_0 \quad \exists x \quad [ |x| = n \land \text{Runtime}_f(x) \geq c_0 \cdot g(n) ]
\]
Now to show $\text{BC}_f(n) = O(g(n))$

we have to show

$\exists n_0, c_0 \forall n \geq n_0 \quad \text{BC}_f(n) \leq c_0 \cdot g(n)$

$\exists n_0, c_0 \forall n \geq n_0 \quad \left[ \min_{x : |x| = n} \{ \text{Runtime}_f(x) \} \leq c_0 \cdot g(n) \right]$}

To show $\text{BC}_f(n) \in \Omega(g(n))$

we have to show

$\exists n_0, c_0 \forall n \geq n_0 \quad \text{BC}_f(n) = c_0 \cdot g(n)$

$\exists n_0, c_0 \forall n \geq n_0 \quad \left[ \min_{x : |x| = n} \{ \text{Runtime}_f(x) \} \geq c_0 \cdot g(n) \right]$}

$\exists n_0, c_0 \forall n \geq n_0 \quad \forall x \left[ |x| = n \Rightarrow \text{Runtime}_f(x) \geq c_0 \cdot g(n) \right]$
\[ WC_f(n) \in O(g(n)) : \]
\[ \exists n_0, c_0 \ \forall n \geq n_0 \ \forall x \ [ |x| = n \ \Rightarrow \ \text{Runtime}_f(x) \leq c_0 \cdot g(n) ] \]

\[ WC_f(n) \in \Omega(g(n)) : \]
\[ \exists n_0, c_0 \ \forall n \geq n_0 \ \exists x \ [ |x| = n \ \land \ \text{Runtime}_f(x) \geq c_0 \cdot g(n) ] \]

\[ BC_f(n) \in O(g(n)) : \]
\[ \exists n_0, c_0 \ \forall n \geq n_0 \ \exists x \ [ |x| = n \ \land \ \text{Runtime}_f(x) \leq c_0 \cdot g(n) ] \]

\[ BC_f(n) \in \Omega(g(n)) : \]
\[ \exists n_0, c_0 \ \forall n \geq n_0 \ \forall x \ [ |x| = n \ \Rightarrow \ \text{Runtime}_f(x) \geq c_0 \cdot g(n) ] \]
Example 1

\textbf{ALL-EVEN} \ (A)

\[ n = \text{len} \ (A) \]

\[
\begin{aligned}
\text{For } i = 1 \text{ to } n \\
\quad \text{If } A[i] = \text{odd} \\
\quad \quad \text{Print } "\text{Not all even}"
\end{aligned}
\]

What is $\text{WC}_{\text{ALL-EVEN}} \ (n)$? $\Theta(n)$

What is $\text{BC}_{\text{ALL-EVEN}} \ (n)$? $\Theta(1)$
Want to show $WC_{\text{ALL-EVEN}}(n) \in \Theta(n)$.

(1.) Show $WC_{\text{ALL-EVEN}}(n) \in O(n)$

Want to show:

$$\exists n_0 \in \mathbb{N} \exists c_0 \in \mathbb{R}^+ \forall n \geq n_0 \forall x$$

$$\text{Runtime}_{\text{ALL-EVEN}}(x) \leq c_0 \cdot n$$

[if $x$ is a list of length $n$]

Let $n_0 = 1$, $c_0 = 1$. Let $n \geq n_0$.

Let $x$ be a list of length $n$.

Have to prove this for all inputs lists of length $\geq n_0$, so use static analysis [analysis depends on $n$ but not on the actual input of length $n$]
We will consider (*) to take one step since it is a constant sized block of code.

I will consider the runtime on $x$ to be the number of executions of the For-loop. Since $x$ has length $n$, the For-loop executes at most $n$ times.
(2.) Show \( WC_{\text{ALL-EVEN}}(n) \in \Omega(n) \)

Want to show

\[
\exists n_0 \in \mathbb{N}, \forall n \geq n_0 \exists x
\]

\[
[x \text{ is a length } n \text{ list and }
\text{Runtime}_{\text{ALL-EVEN}}(x) \geq c_0 \cdot n]
\]

Let \( n_0 = 1 \), \( c_0 = 1 \). Let \( n \geq n_0 \).

Let \( x \) be the following list of length \( n_0 \)

\[
x[i] = 2, \quad \text{for } i = \{2, \ldots, n \}
\]

The runtime of algorithm on this \( x \) is \( n \) since all entries are even, so loop won't quit early, and it will execute \( n \) times.

Here you must construct an input family, one input for every \( n \geq n_0 \).
i. since $\text{WC}_{\text{ALL-EVEN}}(n) \in O(n)$ and $\in \Omega(n)$

it follows that $\text{WC}_{\text{ALL-EVEN}}(n) \in \Theta(n)$
Want to show $BC_{\text{ALL-EVEN}}(n) \in \Theta(1)$

1. $BC_{\text{ALL-EVEN}}(n) \in \Omega(1)$

Show: $\exists \varepsilon \in \mathbb{C} \forall n \geq n_0 \forall \lambda \in A \times$

(If $\lambda$ has length $n$, then $\text{Runtime}_{\text{ALL-EVEN}}(\lambda) \geq \varepsilon$)

Pick $n_0 = \varepsilon_0 = 1$. Let $n = n_0$. Let $x$ be a list of length $n$.

Any algorithm on any input requires at least one step, so on $x$ this algo requires $\geq \varepsilon = 1$ steps.
\[ 2 \text{ Show } \mathcal{B}_\text{ALL-EVEN} (n) \leq O(1) \]

Show:

\[ \exists n_0 \in \mathbb{N}_0 \text{ s.t. } \forall n \geq n_0 \exists x \]

[x is a list of length \( n \) and Runtime \( f_{\text{ALL-EVEN}} (x) \leq c_0 \]

Let \( n_0 = c_0 = 1 \). Let \( n \geq n_0 \).

Let \( x \) be the following input list of length \( n \):

\[ x[i] = 1 \quad i = 1, \ldots, n \]

\( \text{ALL-EVEN} \) on \( x \) will only execute the for-loop once since \( x[1] \) is odd.

So runtime is \( 1 \).
Example 2

Alg-FUNNY (n)

If n is even:
For i = 1 to n^2
Print "Funny"

Else
For i = 1 to n
Print "Not funny"

What is the runtime as a function of n?
For any algorithm with only one input \( n \) of a given size, \( n \):

\[
WC_{\text{Alg}}(n) = \max_{x, |x| = n} \text{Runtime}_{\text{Alg}}(x)
\]

\[
BC_{\text{Alg}}(n) = \min_{x, |x| = n} \text{Runtime}_{\text{Alg}}(x)
\]

So

\[
WC_{\text{Alg}}(n) = BC_{\text{Alg}}(n)
\]

\[
= \text{Runtime}_{\text{Alg}}(n)
\]
Runtime of \textsc{Alg-FUNNY}

Runtime $n^2$

Actual runtime

Runtime $(n) \in \mathcal{O}(n^2)$

\text{Cannot prove a nice $\Theta$ bound for this algorithm}
Show \( \text{Runtime}(n) \in O(n^3) \):

For any \( n \), if the "IF" executes,
\[ \text{# executions of loop} \leq n^2 \]
if "ELSE" executes instead,
\[ \text{# executions of loop} \leq n \]
\[ \implies \text{the total # of steps} \leq n^2 \]

Show \( \text{Runtime}(n) \in \Omega(n^3) \):

show \( \exists \theta \geq 0 \) s.t. \( n \geq n_0 \implies \text{Runtime}(n) \geq \theta \cdot n^3 \)

If the "IF" executes, runtime is \( n^2 \)
otherwise if "ELSE" executes, runtime is \( n \)
so no matter what, runtime \( (n) \geq n \).
$r_k = j_k - i_k$

Just before $k$th iteration of loop

\[ L = \{2, 5, 6, 7, 9, 10, 15, 20\} \]

\[ \text{start: } i_0 = n \]

(b) terminate when $i_k < j_k$

\[ j_k = i_k = 0 \]

$n \rightarrow \frac{n}{2} \rightarrow \frac{n}{4} \rightarrow \frac{n}{8} \rightarrow \ldots$
Average case complexity

Between worst-case and best case complexity

Describes the "typical" or average runtime on inputs of length \( n \)

Average case can be surprising

More difficult to analyze in general than WC or BC complexity
Now we want to consider algorithms that have multiple inputs of length $n$. How to measure runtime?

- $\text{Avg}_{A}(n)$
- $\text{WC}_{A}(n)$
- $\text{BC}_{A}(n)$
To compute the average case runtime of an algorithm, we had to define the set of all inputs of length $n$.

Let $T_n^A = \text{set of all inputs to algorithm } A$ of length $n$.

$$\text{Average}_{A}(n) = \frac{\sum_{x \in T_n^A} \text{Runtime}_{A}(x)}{|T_n^A|}$$
Example 1

Input is a list of \( n \) numbers, where each number in the list is in \( \{1,...,n\} \) and each \( i \in \{1,...,n\} \) occurs exactly once. So the input is a permutation of \( 1,...,n \).

Ex. if \( n=4 \), a possible input is \( 3,1,3,4 \)

\[ 1,2,3,4 \]

**Def** Find1( \( L \) )

\[ n = \text{Len}(L) \]

For \( i = 1 \) to \( n \)

If \( L[i] = 1 \) halt output \( i \)
What is $T_n^{\text{finds}}$?

What is $|T_n^{\text{finds}}|$?

$T_n^{\text{finds}} = \{\text{orderings/permutions of } 1...n\}$

$|T_n^{\text{finds}}| = n! = n \cdot (n-1) \cdot (n-2) \cdot ... \cdot 1$

why?

$\begin{array}{c}
\text{ex} \\
n=3
\end{array}$

$T_3^{\text{finds}} = \{123, 132, 213, 231, 312, 321\}$

$n$ choices

$n-1$ n-2

altogether $n \cdot (n-1) \cdot (n-2) \cdot ... \cdot 1 = n!$
\[ n = 3 \]

\[ T_{\text{Find1}} \] = \begin{bmatrix}
1 \\
123, 132 \\
2213, 2313 \\
2312, 3213
\end{bmatrix}

Avg runtime area \[ T_{\text{Find1}} \]

\[ = \frac{1 \cdot 2 + 2 \cdot 2 + 3 \cdot 2}{6} \]

\[ = 2 \]

Group all of these inputs: \[ T_{1,n}, T_{2,n}, T_{3,n} \]

- \[ T_{1,n} = \text{all inputs with } 1 \text{ in 1st position} \]
  \[ \text{runtime} \]
  \[ \text{runtime} \]

- \[ T_{2,n} = \]

- \[ T_{3,n} = \]

\[ \text{runtime} \]

\[ \text{runtime} \]
Average \( F(n) = \sum_{x \in T} \frac{\text{Runtime}(x)}{n!} \)

Example \( n = 4 \):

\[
\begin{align*}
1 & \times 4 \text{ inputs in T with 1 in 1st location} \\
2 & \times 4 \text{ inputs in T with 1 in 2nd location} \\
3 & \times 4 \text{ inputs in T with 1 in 3rd location} \\
4 & \times 4 \text{ inputs in T with 1 in 4th location}
\end{align*}
\]

\[
\begin{align*}
1 & \times 1234 \\
2 & \times 1243 \\
3 & \times 1342 \\
4 & \times 1432
\end{align*}
\]

\( 4! \)
\[ \frac{3!}{4!} (1+2+3+4) = \frac{1}{4} (1+2+3+4) \]

Now let's do the general case.

Average \( \text{Find}_1(n) = \sum_{x \in T_n} \text{Runtime of } \text{Find}_1 \text{ on } x / n! \)

\[ = \sum_{i=1}^{n} i \cdot \left( \# \text{q's in } T_n \text{ with 1 in location } i \right) / n! \]

\[ = \left[ \sum_{i=1}^{n} i \cdot (n-1) \right] / n! \]

\[ = \frac{(n-1)! \sum_{i=1}^{n} i}{n!} = \frac{1}{n} \left( \frac{n \cdot (n+1)}{2} \right) = \frac{n+1}{2} e \cdot \Theta(n) \]
So what did we do?

We found a way to partition all of the inputs in $T_n$ into disjoint "buckets" or subsets, so that

- all inputs in the same bucket had the same runtime
- every input is in exactly one bucket