Midterm 2 topics
- recursion
  - division-cong
    - backtracking
  - dynamic programming
- graphs
  - traversal/reachability
  - dagst/top sort
  - strong components
  - shortest paths

Verifying shortest path tree
- given pred but no dist
- given dist but no pred

Swedish hackers

1. $2 \times 3 + 0 \times 6 \times 1 + 4 \times 2$

2. Elmo

- Start
- Finish

3. Given $G = (V, E)$ directed weighted edges $l(e)$
   - given $\text{pred}(v)$ for every vertex except one $s$
   - Verify that $\text{pred}$s describe a shortest path tree rooted at $s$

   - pred edges are edges in $G$
     $\text{pred}(v) \rightarrow v \in E$ for all $v$
   - pred edges define a tree!

   - Is every vertex reachable from $s$ through pred edges?

4. Shortest path distances consistent in $T$ and $G$
   - Compute distances in $T$
     $\text{dist}_T(v) = \begin{cases} 0 & \text{if } v = s \\ \text{dist}_T(\text{pred}(v)) + l(\text{pred}(v)s) & \text{otherwise} \end{cases}$
   - $\text{dist}_T(v)$ is an preorder traversal of $T$

   - $O(|V|)$
- Check all edges in $G$ if any tense, fail. $O(E)$

for all vertices $u$
for all edges $u-v$
if $\text{pred}(v) = u$
mark $v$

if any vertex unmarked, fail $+ O(V)$

---

Elmo game

5 6 2 7 1 6 5
Left or right

Elmo game 2

5 6 9 2 7 1 6 5
keep end turn
or pass and go again

BestScore(i, me) = max score I can get from cards i ... n
if I go first if me = True
Elmo
if me = FALSE

We need BestScore(1, False)

Input: $C[1...n]$ of card values

$\text{BestScore}(i, \text{me}) = \begin{cases} 
O & \text{if } i > n \\
\max \left\{ C[i] + \text{BestScore}(i+1, \text{me}) \right\} & \text{if } \text{me} = \text{True} \\
\min \left\{ C[i] + \text{BestScore}(i+1, \text{me}) \right\} & \text{if } \text{me} = \text{False} 
\end{cases}$

$O(n)$ time
Swedish hackers

Given target value $x$:

Find $x$ if $x$ is not corrupted.

$k=1$

- Compare all 3 to $x$
- Follow majority opinion
- Compare median of 3 with $x$

$k>1$

Use a window of size $2k+1$

$O(k \log \frac{1}{k})$

$k$ $\leftarrow$ $k-1$

$O(2^k \log n) \text{ time?}$
Longest increasing path \( O(E) \)
Remove any decreasing edges
Remaining subgraph is a dag
add \( s \to v \) to every \( v \to t \) from every \( v \)

Run the longest path algo
from class \( O(V+E) \) time
\( = O(V+E) \)

Given an expression like this: 
\[ (2 \times 3) + (0 \times 6) \times 1 + (4 \times 2) = 14 \]
Find min value we can get by inserting parens.
Assume input \( X[0..2n] \)

Odd positions: 
+ or \( X \)

Even positions: numbers

Binary tree

\[ (((# + #) \times #) + (# \times (\# + (# \times #)))) \]
MinValue(i,k) = smallest value we can get
From X[2i...2k] by adding parens
We need MinValue(0,n)

$$\text{MinValue}(i,k) = \begin{cases} 
X[2i] & \text{if } i = k \\
\min_{i \leq j < k} \left\{ \text{MinValue}(i,j) + \text{MinValue}(j+1,k) \right\} & \text{if } X[2j+1] = + \\
\text{MinValue}(i,j) \times \text{MinValue}(j+1,k) & \text{if } X[2j+1] = \times 
\end{cases}$$

\[O(n^3)\] time

For i ≤ n down to 1
For k = i to n
Obtuse angle maze

Given undirected graph \( G = (V, E) \)

cords for every vertex
every edge is a straight line segment

Two verts start and finish

Find shortest walk in \( G \) from \( s \) to \( f \)

that only makes obtuse turns.

Build a new graph \( G' = (V', E') \)

\[
V' = \{ (u, v) \mid uv \text{ is an edge} \} \\
U \{ s' \}
\]

\[
E' = \{ (u, v) \rightarrow (v, w) \mid uvw \text{ is obtuse} \} \\
U \{ s' \rightarrow (s, v) \mid s u \text{ is edge} \} \quad O(1) \text{ time}
\]

We need shortest path in \( G' \) from \( s' \) to

any vertec \( (., \text{finish}) \)

**BFS at \( s' \)**

time \( = O(V' + E') \)

\( = O(VE) \)

\(\text{Crude}\)

\[
E' \leq \sum \deg(w)^2 = O(VE) \\
\leq \sum V \cdot \deg(w) = V \cdot \sum \deg(w) \\
= O(V \cdot E)
\]

\# obtuse = 7.12
Longest Palindrome in a DAG

1. Top sort G

\[ LPP(u, x) = \begin{cases} \max \# \text{nodes} \\ \text{length of longest pal. path} \end{cases} \]

\[ \text{starts at } u \quad \text{ends at } x \]

Add source \( s \) and sink \( t \)

\[ s \rightarrow v \]

\[ v \rightarrow t \]

Now we need \( LPP(s, t) - 2 \)

\[ LPP(u, x) = \begin{cases} \infty & \text{if } \text{label}(u) \neq \text{label}(x) \\ 1 & \text{if } u = x \\ \max \{ LPP(v, w) + 2 \mid u \rightarrow v, v \rightarrow w, w \rightarrow x \in E \} \quad \text{(also 2 if } u \rightarrow x \text{ is edge)} \end{cases} \]

For all verts \( u \) in rev top order

For all verts \( x \) in top order

\[ \Theta(V^4) \]

\[ O(E^2) \text{ time} = O(V^4) \]