Dynamic Programming
Randomized Algs/DS
Flows, Cuts, LP
NP-hardness, Approx Algs

Q: Can you win?  □ = TRUE  □ = FALSE

Time: $O(4^d) = O(n)$

1. Prove you can't avoid worst-case: every leaf
2. Randomize! $O(c^d)$ exp. time for some $c < 4$.
   Postorder, but randomly choose first child at every node

$T(2d) \leq 3 \cdot T(2d-2)$

$\Rightarrow O(3^d)$ exp. time.
\[
\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2 = \frac{3}{2}
\]

2 recursive calls

\[
E[\#\text{calls}] = \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 3 = \frac{8}{3}
\]

\[
\Rightarrow T(d) \leq \frac{8}{3} \cdot T(d-1) = O(\left(\frac{8}{3}\right)^d)
\]
7 2 3 4 1 6 5
7 2 3 5 6 1 4 3
7 2 5 6 3 4 1

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

burned
pancake
sorting
Gates + Papadimitriou
Input: $A[1..n, 1..4]$

Choose subset with max sum, but forbidding adjacent pairs

Max Sum($i, b$) = maximum sum from rows $i$ onward where row $i-1$ uses bit pattern $b$.

Max Sum($i, b$) = $\max_{b'\text{ consistent with } b} (\sum_{j=1}^{i} b'[j] A[i, j] + \text{MaxSum}(i, 1))$

Initial call: MaxSum(1, 0000)

Data structure: $n \times 2^k$ array

$\mathcal{O}(1)$

$\mathcal{O}(n)$ time
Bus home from Siebel

List of bus routes:
- Stops and times
- Length N

Minimize time waiting for buses out in the rain

Build a graph:
- \( V = \{ \text{stop, time} \} \) for \( |V| = N \)
- \( E = \{ (\text{stop}, \text{time}) \rightarrow (\text{stop}, \text{time}) \} \) for all successive steps
- \( U = \{ (\text{stop}, \text{time}) \rightarrow (\text{stop}, \text{time}) \} \) for some buses

Shortest path from (Siebel, time) to (home, time)

Dijkstra: \( O(E \log V) = O(N \log N) \)

DAG: DFS/DP \( O(E) = O(N) \)

\( \text{longest}(v) = \max \text{ length } v \rightarrow t \)

- \( \sum D \) if \( v = t \)
- \( \infty \) if \( v \neq t \) and \( v \text{ sink} \)
- \( \max (llv \rightarrow w) + \text{longest}(w) \)
Route calls $3 \rightarrow 7$

$5 \rightarrow 4$  $\vdots$

To minimize congestion

$\max_e (\# \text{calls thru } e)$

Problem: Find a $2$-approx algorithm

$A(x) \leq f(x) \leq 2 \cdot \text{OPT}(x)$

We know $A(x) \geq \text{OPT}(x)$

$\text{OPT}(x) \geq \sum_{\text{call } i} \frac{\text{length}(i)}{\text{call } i} \cdot \frac{1}{n}$

Guess algo: Route X-ockwise.

Either $a_1$ or $c_2$, whichever is better shortest

For each call, route to increase as little as possible.
Pack boxes into bins minimize \# bins

"Knapsack problem"

\[ A(x) = f(x) \leq Z \cdot \text{OPT}(x) \]

\[ \text{OPT}(x) = \left\lceil \frac{\sum W_i / W_{\text{max}}}{Z} \right\rceil \]

Sort by decreasing weight

**Best Fit:** For \( i = 1 \) to \( n \)
- put item \( i \) into emptiest nonempty bin or add bin if item \( i \) doesn't fit...

**First Fit:**
- \( j = 1 \)
- for \( i = 1 \) to \( n \)
  - if item \( i \) fits
    - put in \( j \)
  - else
    - \( j = j + 1 \)
    - put \( i \) into \( j \)

**Claim:** At most one bin is less than half full.

\[ A(x) \leq 1 + \frac{1}{2} Z \cdot \frac{\sum W_i / W_{\text{max}}}{Z} \leq \frac{3}{2} \text{OPT}(x) \]
For all edges $e$:

IF $TSP\text{Length}(G) = TSP\text{Length}(G-e)$

$G = G - e$

return $G$
Sway Partition \( X \rightarrow A \cup B \cup C \)

\[ \Sigma A = \Sigma B = \Sigma C \]

\[ Y = X \cup \frac{\Sigma X}{2} \]

**Proof:**

1. **Reduction**
   - Proof
   - \( L = 3 + 3 \)

2. **Proof**
   - Suppose \( X \) can be partitioned
     - \( X = A \cup B \) where \( \Sigma A = \Sigma B \)
     - Then \( \Sigma A = \Sigma B = \Sigma X/2 \)
     - So \( Y \) can be 3-partitioned.

3. **(\Rightarrow) Suppose** \( Y \) can be 3-partitioned
   - \( Y = A \cup B \cup C \)
   - where \( \Sigma A = \Sigma B = \Sigma C = \Sigma Y/3 \)
   - So \( \Sigma A = \Sigma B = \Sigma C = \Sigma X/2 \)
   - \( \Rightarrow A \cup B = X \)
   - So \( X \) can be partitioned. \( \square \)
Box Depth $\rightarrow$ Clique

Prove $X$ is NP hard
Cycle Cover

Input: directed $G = (V,E)$
Output: $T/F$

Collection of cycles in $G$ touching each vertex once

For every node $v$ assign a node $\text{next}(v)$

Build bipartite graph $H$:
- $V(H) = V^+ U V^-$
- $E(H) = \{uv \mid u^+ v^- \in E\}$

Find perfect matching in $H$.

IF $H$ has perfect matching
return $T$
else
return $F$

$O(V,E)$ time