CompSci 260P
Spring 2023 Lecture:
Review of Diagnostic Exams
Diagnostic 1, Dynamic Programming

- Word Segmentation: Maximize sum of quality
- English definition then clear recursive formula

\[
\sum_{i=1}^{n} \text{Segment}(i) = 0
\]

\[
\text{def Segment}(i) \text{ as highest quality break-up for } s[1...i]
\]

\[
\max \{ \text{Segment}(j-1) + \text{quality}(s, j, i) \mid 1 \leq j \leq i \}
\]

\[O(n) \text{ work/case } \times O(n) \text{ cases: } \Theta(n^2)\]

Memoize vector [0...n], increasing index
Diagnostic 2, Dynamic Programming

- Candy Jar
  - $n$ office hours
  - Jar holds $C$ pieces of candy
  - Costs $P$ to refill jar
  - Costs $d$ per piece after each office hour
  - Office hour $i$ expects $s_i$ students

- Be sure to understand the problem!
Vector \([0 \ldots n, 0 \ldots C]\), \(O(nC)\) cases, \(O(1)\) time each

Candy Jar incr both dims Goal: Candy\((n, 0)\) total

English definition then clear recursive formula

Let Candy\((i, j)\) be least expensive way
for OTs \(1 \ldots i\) w/ \(j\) leftover after \(i\)th.

\[
\text{let } \text{Candy}(i, j) = \begin{cases} 
\text{Candy}(i-1, 0) + p, & \text{if } j + S_i > C \\
\text{Candy}(i-1, j + S_i) + d(j + S_i), & \text{otherwise}
\end{cases}
\]

base? Candy\((0, *) = 0\)
Diagnostic 1, Complexity

▶ Trucks! Version One:
  ▶ Constraints are canister to truck.
  ▶ Each truck has max carrying capacity
  ▶ Load all trucks?

▶ Version Two
  ▶ Constraints are canister to canister
  ▶ Each truck has max carrying capacity
  ▶ Load all trucks

▶ Polynomial time Solvable?
Diagnostic 1, Complexity

- Version Two is $\mathcal{NP}$-complete.
  - Constraints are canister to canister
  - Each truck has max carrying capacity
  - Load all trucks

3-COLOR ($G$)

\[
\begin{align*}
\text{for each vertex } i & : \\
\text{def canister } i \\
\text{for each edge } (i,j) & : \\
\text{constraint truck } (i) & \neq \text{truck } (j)
\end{align*}
\]

$V_2$ (canisters, 3 trucks, $|V_1|$ capacity)
Diagnostic 2, Complexity

- *Spacing*: min dist btwn vertices in diff sets
- *Diameter*: max dist btwn vertices in same set
- Polynomial time Solvable?
Diagnostic 2, Complexity

- **Spacing**: min dist btwn vertices in diff sets
- **Diameter**: max dist btwn vertices in same set
- $\mathcal{NP}$-complete?

3-color $(h)$