Dynamic Programming

- Like Divide and Conquer, build solution to big problem from smaller
- Unlike "", save all previous solutions in memory (like HW problem to find 2nd largest element of array)
- Easiest to see an example...

Max-Weight Independent Set Problem (MWISP)

Input: Graph \((V, E)\) and weight function \(w: V \rightarrow \mathbb{R}^+\)

Output: \(S \subseteq V\) s.t. if \((v_i, v_j) \in E\), \(v_i, v_j\) can't both be in \(S\).

This set is Max Weight Ind. Set (MWIS)

\[
\sum_{v \in S} w(v) \text{ is max over all possible independent sets}\]

\[w(S)\] "Objective function"

Applications

- WiFi transmitters / cell towers

- Tower \(i\) has \(n(i)\) packets
- If 2 towers are \(\leq d\) distance \(d\), causes interference if both transmit

Q: How to use MWIS to figure out which towers should transmit?
**MWISP on Path Graph**

\[ V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow \ldots \rightarrow V_n \]

\[ S \rightarrow 7 \rightarrow 6 \rightarrow 1 \]

*Can create a divide & conquer alg., but performance not optimal*

**Instead:**

1. Consider form of optimal solution \( S \leftarrow MWIS \)

\[ (\bigcirc \ldots \bigcirc) \leftarrow \text{last vertex has two options} \]

- \( i) \) \( v_n \notin S \)
- \( ii) \) \( v_n \in S \)

i) If \( v_n \notin S \), \( S \) is MWIS on \( G_{n-1} \)

\[ \text{Pf:} S \text{ is an ind. set of } G_{n-1} \]

- \( S \) must be max-weight ind. set of \( G_{n-1} \) (otherwise, choose better set \( S' \); \( S' \) also better than \( S \) on \( G_n \Rightarrow \text{contradiction} \))
2. How is $S$ related to solution of smaller problem

\[ \text{last vertex has two options} \]
\[ \text{i) } v_n \notin S \]
\[ \text{ii) } v_n \in S \]

i) If $v_n \notin S$, $S$ is MWIS on $G_{n-1}$

**Pf:** $S$ is an ind. set of $G_{n-1}$

- $S$ must be max-weight ind. set of $G_{n-1}$ (otherwise, choose better set $S'$, $S'$ also better than $S$ on $G_n \Rightarrow$ contradiction)

ii) $v_n \in S$, $S - v_n$ is MWIS on $G_{n-2}$

**Pf:** $S - v_n$ is a valid ind. set for $G_{n-2}$

- $S_n - v_n$ must be max weight ind. set on $G_{n-2}$ (If $S'$ better, $S' \cup v_n$ is better ind. set for $G_n \Rightarrow$ contradiction)
Conclusion:

MWIS on $G_n$ is

i) MWIS on $G_{n-1}$

OR

ii) MWIS on $G_{n-2} + v_n$

\[
\begin{array}{c}
G_1 \\
| \quad | \\
G_{n-1} \\ G_{n-2}
\end{array}
\]

Take max of these two options!

\[
A) O(1) \quad B) O(n) \quad C) O(n^2) \quad D) O(2^n)
\]

Levels = $O(n)$

#subproblems double at each level

This is bad! Work just in base case is $O(2^n)$!
But, let's look more carefully at recursive calls.

Actually solving same problem over and over!

Q. How many distinct subproblems are there?

A) $O(1)$  B) $O(n)$  C) $O(n^2)$  D) $O(2^n)$

\[
\{G_1, G_2, \ldots, G_n\}
\]

3. Create recurrence relating value of optimal solution to smaller solutions.