S.KIMMEL

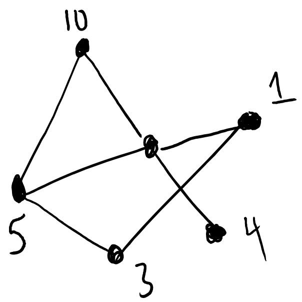
Dynamic Programming

- Like Divide and Conquer, build solution to big problem from smaller
- Unlike " ", save all previous solutions in memory

(like HW problem to find 2^{nd} largest element of array)

- Easiest to see an example...

Max-Weight Independent Set Problem (MWISP)



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

vertices edges

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

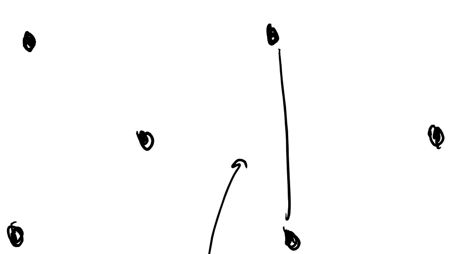
• $\sum_{v \in S} w(v)$ is max over all possible independent sets } max weight

$\sum_{v \in S} w(v) = w(S)$ ← "Objective function"

This set is Max Weight Ind. Set (MWIS)

Applications

• Wifi transmitters / cell towers



weight is amount of data needs to send

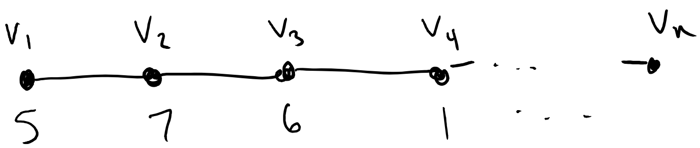
connect if within distance d.

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

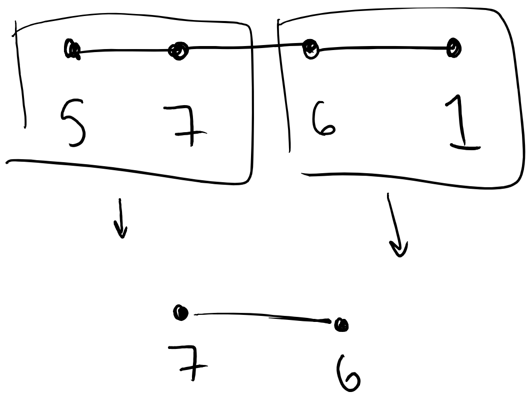
Q: How to use MWIS to figure out which towers should transmit?

S.KIMMEL

MWISP on Path Graph



Divide & Conquer

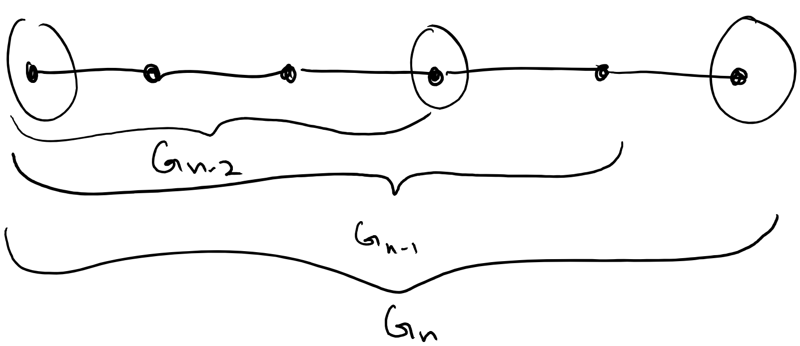


can't combine!

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

Instead:

1. Consider form of optimal solution $S \leftarrow$ MWIS



$\leftarrow$ last vertex has two options

i) $v_n \notin S$

ii) $v_n \in S$

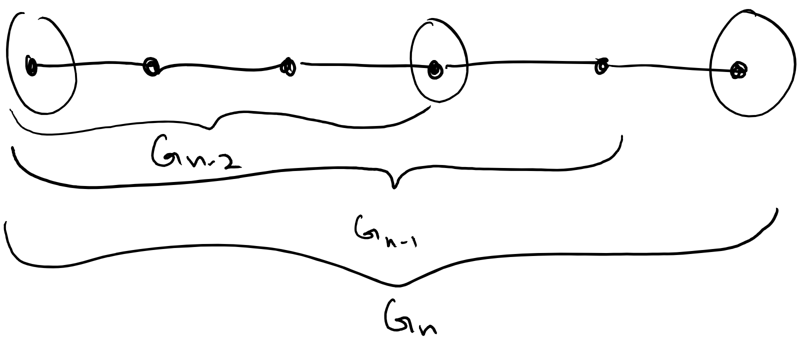
i) If $v_n \notin S$. S is MWIS on G_{n-1}

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)

S.KIMMEL

2. How is S related to solution of smaller problem



← 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}

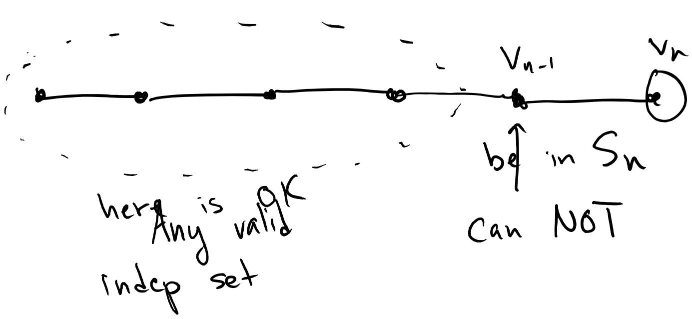
Pf: • S is an ind. set of G_{n-1}

• 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 → contradiction)



S. KIMMEL

Conclusion:

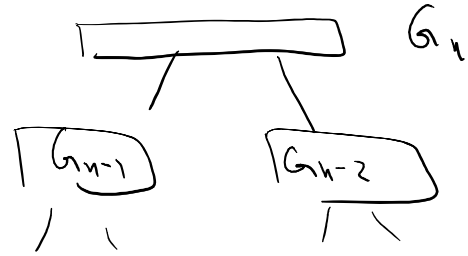
MWIS on G_n is

i) MWIS on G_{n-1}

OR

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

Take max of these two options!



← How many subproblems at base case?

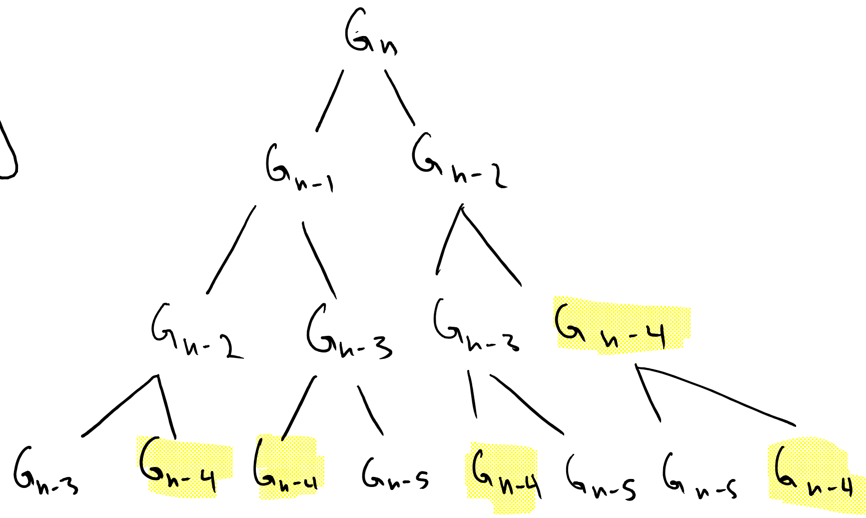
A) $O(1)$ B) $O(n)$ C) $O(n^2)$ D) $\boxed{O(2^n)}$

Levels = $O(n)$
# subproblems double at each level

This is bad! Work just in base case is $O(2^n)$!

S.KIMMEL

But, let's look more carefully at recursive calls



$O(n)$ levels, each level doubles # of 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) $O(2^n)$

$\{G_1, G_2, \ldots G_n\}$

$\boxed{3.}$ Create recurence relating value of optimal solution to smaller solutions.