

VC – Recursively

Vertex Cover

A set S of vertices is a vertex cover if for every edge (u, v) : u is in S , or v is in S , (or both)

Let's try to write a recursive algorithm first.

What information do we need to decide if we include u ?

If we don't include u then to be a valid vertex cover we need...

to include **all** of u 's children, and vertex covers for each subtree

If we do include u then to be a valid vertex cover we need...

just vertex covers in each subtree (whether children included or not)

52

Vertex Cover Dynamic Program

What memoization structure should we use?

What code should we write?

What's the running time?

8

Ordering

Instead of $dist(v)$, (the true distance) right from the start, we'll let $dist(v, i)$ to be the length of the shortest path from the source to v that uses at most i edges.

That breaks ties – counting the number of edges required!

$$dist(v, i) =$$

21

Another Recurrence

Put the vertices in some (arbitrary) order $1, 2, \dots, n$

Let $dist(u, v, i)$ be the distance from u to v where the only intermediate nodes are $1, 2, \dots, i$

$$dist(u, v, i) = \begin{cases} ? & \text{if } i = 0, (u, v) \text{ exists} \\ ? & \text{if } i = 0, u = v \\ ? & \text{if } i = 0, \text{ no edge } (u, v) \\ ? & \text{otherwise} \end{cases}$$

53