Vertex Cover – Recursively

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…

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

Recurrence

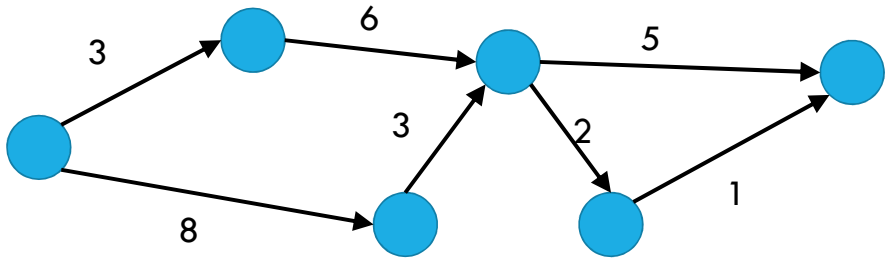
Let $OPT(v)$ be the weight of a minimum weight vertex cover for the subtree rooted at v.

Write a recurrence for $OPT()$

Then figure out how to calculate it

In a DAG

$$dist(v) = \begin{cases} 0 & \text{if } v \text{ is the source} \\ \min\limits_{u:(u,v)\in E}\{dist(u) + weight(u,v)\} & \text{otherwise} \end{cases}$$



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!

$dist(v, i) =$