

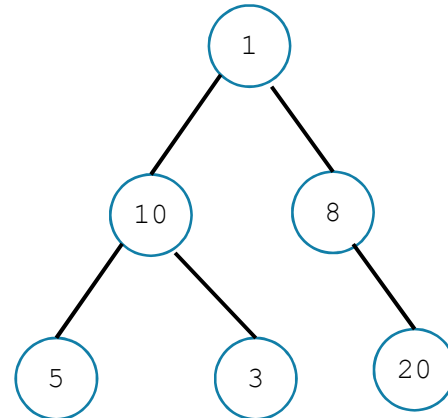
Vertex Cover

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)

Find the minimum vertex cover in a tree.

Give every **vertex** a weight, find the minimum weight vertex cover



Vertex Cover – Recursively

Let's try to write a recursive algorithm first.

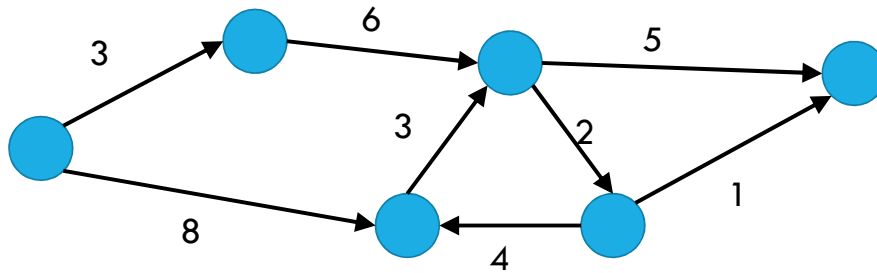
What information do we need to decide if we include u ?

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...

What about cycles?

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



Laundry List of shortest pairs (so far)

Algorithm	Running Time	Special Case only	Negative edges?
BFS	$O(m + n)$	ONLY unweighted graphs	X
Simple DP	$O(m + n)$	ONLY for DAGs	X
Dijkstra's	$O(m + n \log n)$		X
Bellman-Ford	$O(mn)$		Yes!