Longest Increasing Subsequence

$LIS(i, j)$ is "Number of elements of the maximum increasing subsequence from $0, \ldots, i$ where every element of the sequence is at most $A[j]$"

Need a recurrence

$$LIS(i,j) = \begin{cases} 0 & \text{if } i < 0 \\ \mathbb{I}[A[i] \le A[j]] & \text{if } i = 0 \\ LIS(i-1, j) & \text{if } A[i] > A[j] \\ \max\{1 + LIS(i-1, i), LIS(i-1, j)\} & \text{otherwise} \end{cases}$$

If $A[i] > A[j]$ element i cannot be included in an increasing subsequence where every element is at most $A[j]$. So taking the largest among the first $i - 1$ suffices.

If $A[i] \le A[j]$, then if we include i, we may include elements to the left only if they are less than $A[i]$ (since $A[i]$ will now be the last, and therefore largest, of elements $0 \ldots i$. If we don't include i we want the maximum increasing subsequence among $0 \ldots i - 1$.

LIS

One more thing….what's the final answer?

We want the longest increasing sequence in the whole array.

$LIS(i, j)$ is "Number of elements of the maximum increasing subsequence from $0, \ldots, i$ where every element of the sequence is at most $A[j]$"

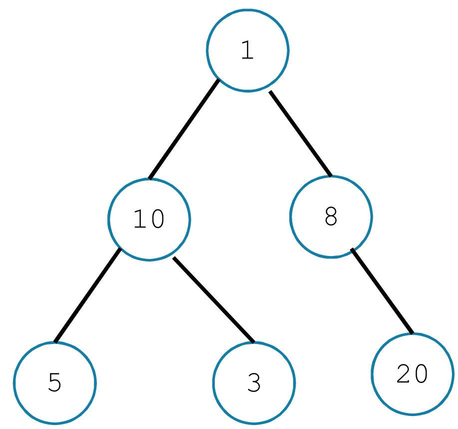
What do we want?

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



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