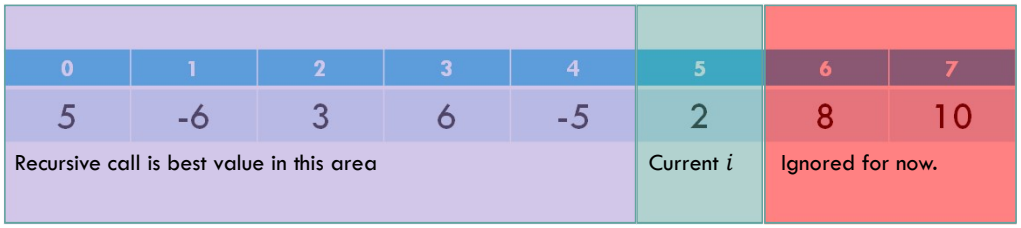
# Recurrence




Need recursive answer to the left

Currently processing $i$

Recursive calls to the left are needed to know optimum from $1 \ldots i$

Will move $i$ to the right in our iterative algorithm

# 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

$$LIS(i,j) = \begin{cases} ? & \text{if } i < 0 \\ ? & \text{if } i = 0 \\ ? & \text{if } A[i] > A[j] \\ ? & \text{otherwise} \end{cases}$$

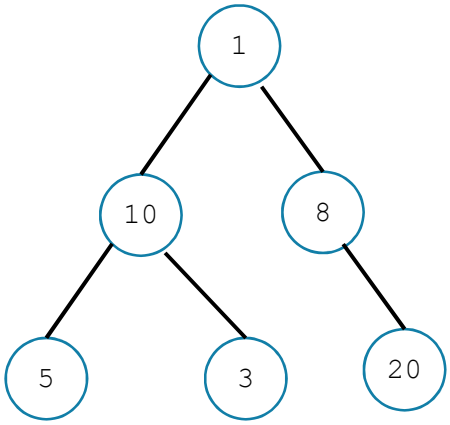
# Vertex Cover

**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