

Two Values

Pollev.com/robbie

Need two recursive values:

INCLUDE(i): sum of the maximum sum subarray among elements from 0 to i that includes index i in the sum

OPT(i): sum of the maximum sum subarray among elements 0 to i (that might or might not include i)

How can you calculate these values? Try to write recurrence(s), then think about memoization and running time.

Example

A	0	1	2	3	4	5	6	7
	5	-6	3	4	-5	2	2	4

$OPT(i)$	0	1	2	3	4	5	6	7
	5							

$INCLUDE(i)$	0	1	2	3	4	5	6	7
	5							

Longest Increasing Subsequence

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

Need a recurrence

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

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 $A[i] \leq A[j]$, 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 $1 \dots i$). If we don't include i we want the maximum increasing subsequence among $1 \dots i - 1$.

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 $0, \dots, i$ where every element of the sequence is at most $A[j]$ "

What do we want?