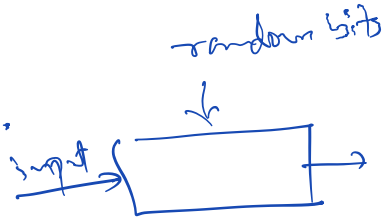
① Quick Sort? – choose last element as pivot.

Worst-case running time – $\Theta(n^2)$.

② Randomized Quick Sort.



E[running time] $= O(n \log n)$

③ Choose **median** as pivot element.

[$T(n) = 2T(n/2) + \Theta(n)$

Worst-case running time $= O(n \log n)$]

Average-case running time ✓

⊗ We need to assume a probability distribution over the inputs.

$\{a_1, a_2 \; - \; - \; - \; a_n\}$.

$a_1 \; - \; - \; - \; a_n$

$a_n \; - \; - \; - \; a_1$ } $n!$

$E[\text{running time}] = O(n \log n)$.

Back-tracking / Branching.

SUBSET-SUM

Input: A set X of n positive integers and a number T

Output: Yes if there is a subset that adds to T.
Otherwise No.

Example

$X = \{2, 4, 8, 10, 1, 12\}$

$T = 18$ — Yes.

$T = 40$ — No

$T = 0$ — yes.

I/p: $x[1 \ldots n]$, T

If the answer is yes, then
- either there is a subset containing $x[n]$ that adds to T
- or there is subset in $x[1 \ldots n-1]$ that adds to T

$$SS(x[1 \ldots n], T) = \begin{cases} \text{yes} & \text{if } T = 0 \\ \text{No} & \text{if } T < 0 \\ \text{No} & \text{if } T > 0, x = \emptyset \\ SS(x[1 \ldots n-1], T) \text{ or} \\ \quad \vee \\ SS(x[1 \ldots n-1], T - x[n]) \end{cases}$$

SS
SUBSET-SUM $(x[1 \ldots n], T)$

(1) Base case.

(2) $a \leftarrow SS(X[1..n-1], T)$

$b \leftarrow SS(X[1..n-1], T - X[n])$.

Return $a \vee b$.

$$T(n) = 2T(n-1) + 5$$

$$T(0) = 1$$

$$T(n) = O(2^n) \quad \checkmark$$

Text segmentation.

Input: A sequence of letters.

Output: Yes if it can be split into words. Otherwise NO.

Example: IAMASTUDENT

I AM A STUDENT

Input: $A[1 \ldots n]$.

function IsWord(⟶)

IsWord(i, j) = Yes if $A[i \ldots j]$ is a proper word.

Splitable(i) = Yes if $A[i \ldots n]$ can be split into words.

if $i > n$

Yes

$$\text{Splitable}(i) = \bigvee_{j=i}^{n} \left[\text{IsWord}(i,j) \wedge \text{Splitable}(j+1) \right]$$

Ans: Splitable(1)?

$$T(n) = \sum_{i=1}^{n} T(n-1) + cn$$

$$T(0) = 1$$

$$T(n) = \sum_{i=1}^{n-1} T(i) + cn$$

$$T(n-1) = \sum_{i=1}^{n-2} T(i) + c(n-1)$$

$$T(n) - T(n-1) = T(n-1) + c$$

$$T(n) = 2T(n-1) + c.$$

$$T(n) = O(2^n).$$

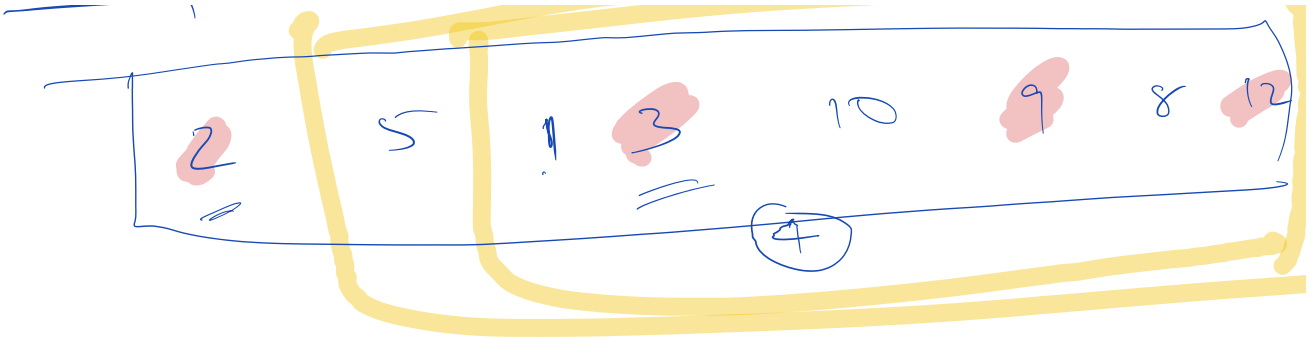
Longest Increasing Subsequence

Input: $A[1 \ldots n]$

Output: Length of a longest increasing subsequence in $A[1 \ldots n]$.

(5)

Example: 10, 2, 4, 8, 12, 22



$LIS(i)$ = the length of a longest increasing subsequence in $A[i \ldots n]$.

$$LIS(i) = \begin{cases} \\ \\ \\ \\ \\ \\ \\ \end{cases} LIS(i+1)$$

We can't write $LIS(i)$ as a function of $\{LIS(j) : j > i\}$.

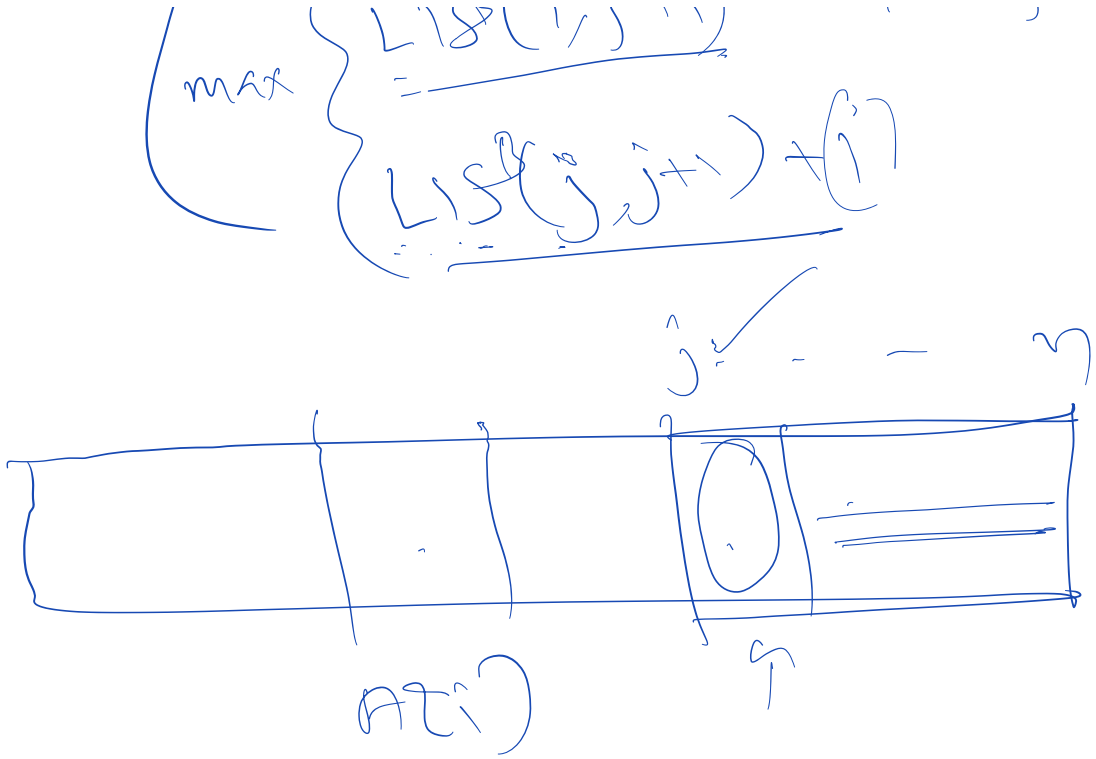
$A[1 \; - \; - \; - \; n]$.

$A[0, 1 \; - \; - \; - \; - \; n]$

For $i < j$,

$LISB(i,j)$ = Then length of a longest increasing subsequence in $A[j \; - \; - \; n]$ that are greater than $A[i]$.

$$LISB(i,j) = \begin{cases} 0 & \text{if } i \geq j \\ LISB(i, j+1) & \text{if } A[j] < A[i] \\ \; & \text{if } A[j] > A[i] \end{cases}$$



Our Answer

LISB(0, 1) :