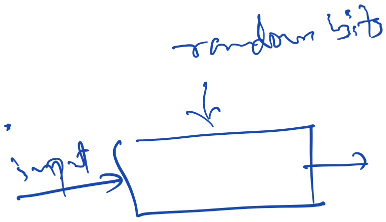
① Quick Sort? — choose last element as pivot.

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

② Randomized Quick Sort.



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

③ Choose median as pivot element.

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

<u>Worst-case running</u> time $= O(n \log n)$]

<u>Average-case running time</u> ✓

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

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

$$\left.\begin{matrix} a_1 - - - a_n \\ a_n - - - a_1 \end{matrix}\right\} n!$$

:)

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

Backtracking | 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

$$SS(X[1 \ldots n], T) = \begin{cases} \text{Yes} & \text{if } T = 0 \\ \text{No} & \text{if } T < 0 \\ \text{No} & \text{if } T > 0, X = \emptyset \\ SS(X[1 \ldots n-1], T) \vee SS(X[1 \ldots n-1], T - X[n]) & \text{o.w.} \end{cases}$$

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

(1) Base case.

(2) $\underline{a} \leftarrow SS(X[1 \ldots n-1], T)$

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

Return $a \vee b$.

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

$$T(0) = 1$$

$$\underline{T(n) = O(2^n)}$$ ✓

<u>Text segmentation</u>.

Input: A sequence of letters.

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

<u>Example</u>: IAMASTUDENT

I AM A STUDENT

Input: $A[1 - - - - n]$.

function IsWord(______)

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

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

$$\text{Splitable}(i) = \begin{cases} \text{Yes} & \text{if } i > n \\ \bigvee_{j=i}^{n} \left[\text{IsWord}(i, j) \wedge \text{Splitable}(j+1) \right] \end{cases}$$

Ans: Splitable(1)?.

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

$T(0) = 1$)

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

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

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

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

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

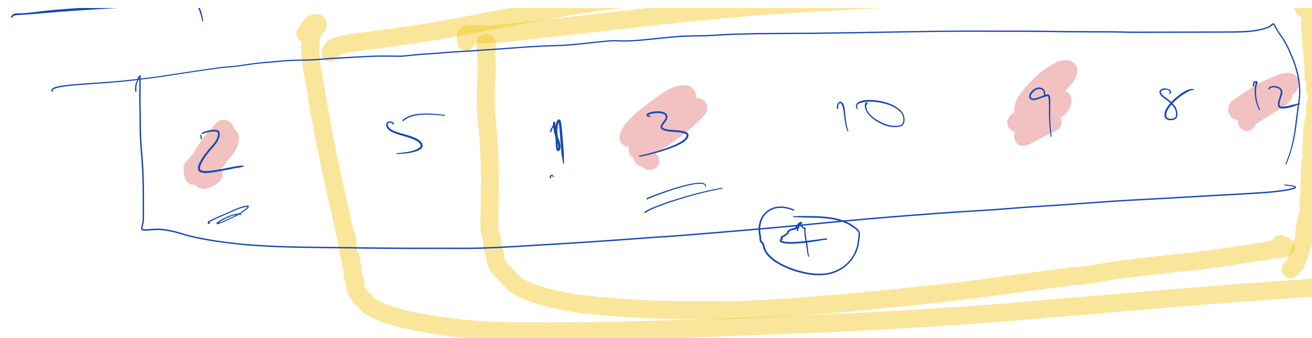
Longest Increasing Subsequence

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...n].

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

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

$A[1 - - - n]$.

$-\infty$, $- - - - -$

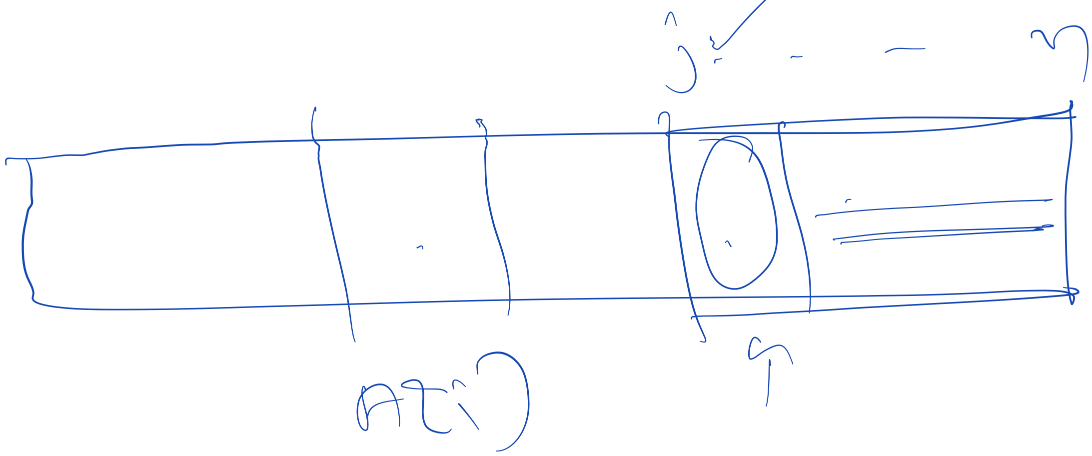
$A[0, 1 - - - - - - n]$

~~if~~ For $i < j$,

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

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

$$\max \begin{cases} LIS(i, j+1) \\ LIS(j, j+1) + 1 \end{cases}$$



Our Question

$LIS(0, 1)$