

Logistics: Masks are recommended.

- HW D due tonight 9pm
- Gradescope self-enrollment code: WVVEEG
- "Describe an algorithm" includes justification but NOT CODE
- "Analyze an algorithm" includes justification

"just"
"simply"
"clearly"
"obviously"

Given a string, can we split it into a sequence of "words"

↑
IsWORD(str) → bool

PRIMVS | DIGNITAS IN TAM TEN VISCIENTIA NON POTEST
ESSERE SENIM SVNT PARVAE PROPE IN SINGVLIS LITTERIS
ATQVE INTERPVNCTIONIBUS VERBORVM OCCVPATAE

Sequence of letters → sequence of words

= { nothing, word }, seq of words
My job → word recursion fairy

What is the first word?

try all prefixes:
IsWORD = TRUE?
Is rest splittable into words?

SPLITTABLE(A[1..n]):

if $n = 0$

return TRUE

for $i \leftarrow 1$ to n

if ISWORD(A[1..i])

if SPLITTABLE(A[i+1..n])

return TRUE

return FALSE

Treat input string A as a global variable

$\text{Splittable}(i) = \text{True}$ iff $A[i..n]$ is splittable into words

$$\text{Splittable}(i) = \begin{cases} \text{TRUE} & \text{if } i > n \\ \bigvee_{j=i}^n (\text{ISWORD}(i, j) \wedge \text{Splittable}(j+1)) & \text{otherwise} \end{cases}$$

```
⟨⟨Is the suffix A[i..n] Splittable?⟩⟩
SPLITTABLE(i):
  if i > n
    return TRUE
  for j ← i to n
    if ISWORD(i, j)
      if SPLITTABLE(j + 1)
        return TRUE
  return FALSE
```

$$T(n) = \text{time for SPLITTABLE}(n) = O(2^n)$$

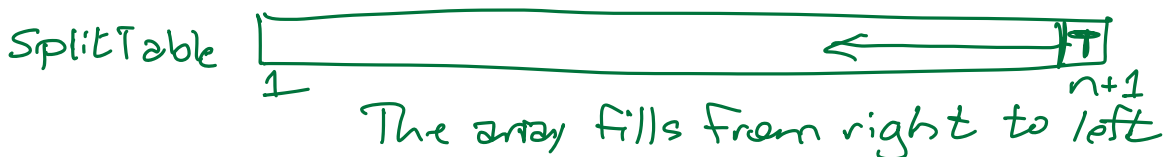
$$T(n) = \sum_{i=1}^n (O(1) + T(n-i))$$

NOWHERE ~~EX~~ QF VY.....

A LOT of redundant computation
MEMOIZATION

Store answers in an array SplitTable [1..n]

Time $\rightarrow O(n^2)$ calls to IsWord



Do that on purpose!

DPSplitTable(A(1..n)):

SplitTable[n+1] ← TRUE

→ For i ← n down to 1
SplitTable[i] ← FALSE

→ For j ← i to n
if iswizd(i, j)
if SplitTable[j+1]
SplitTable[i] ← TRUE

return SplitTable[1]



Longest Increasing Subsequence

What is the first element of LIS?

Is A[1] in the LIS?

Increasing sequence bigger than x

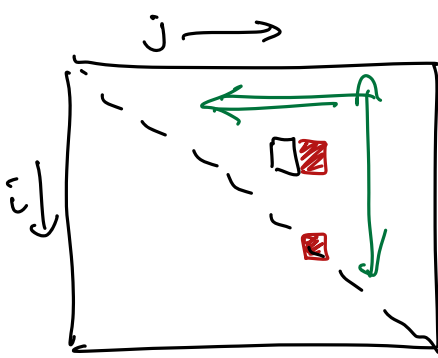
is either \emptyset

or (int y > x) followed by (inc seq > y)

Global variable input A[1..n]

LIS(i, j) = Length of longest inc. subseq. of A[j..n]
bigger than A[i]

$$\text{LIS}(i, j) = \begin{cases} 0 & \text{if } j > n \\ \text{LIS}(i, j+1) & \text{if } A[i] \geq A[j] \\ \max\{\text{LIS}(i, j+1), 1 + \text{LIS}(j, j+1)\} & \text{otherwise} \end{cases}$$



$O(n^2)$ time

FASTLIS(A[1..n]):

$A[0] \leftarrow -\infty$ ⟨⟨Add a sentinel⟩⟩

for $i \leftarrow 0$ to n ⟨⟨Base cases⟩⟩

$LISbigger[i, n+1] \leftarrow 0$

for $j \leftarrow n$ down to 1

for $i \leftarrow 0$ to $j-1$ ⟨⟨...or whatever⟩⟩

$keep \leftarrow 1 + LISbigger[j, j+1]$

$skip \leftarrow LISbigger[i, j+1]$

if $A[i] \geq A[j]$

$LISbigger[i, j] \leftarrow skip$

else

$LISbigger[i, j] \leftarrow \max\{keep, skip\}$

return $LISbigger[0, 1]$

