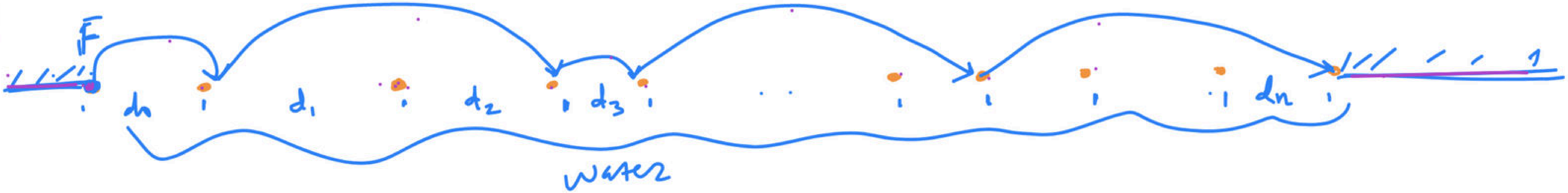
Greedy Algorithms

Course on Game Theory and Greedy Algorithms

Triveni Mahatha • Lesson 2 • May 15, 2021



Max Jump : 'k' units

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

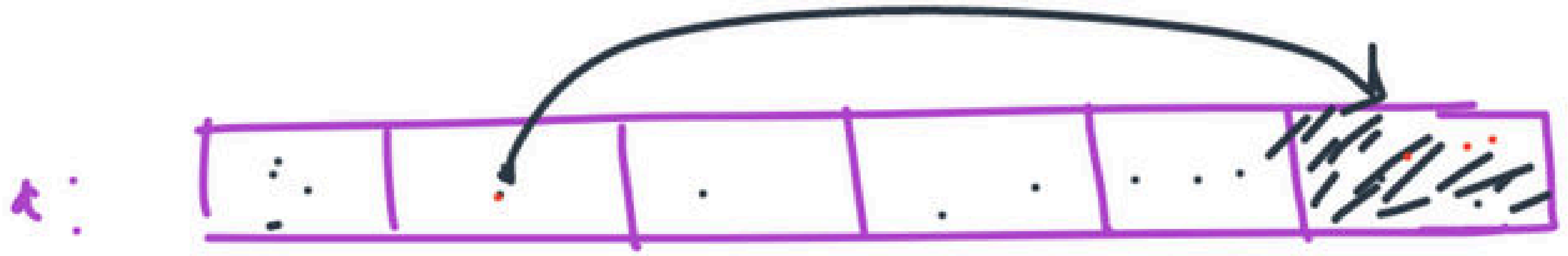
$dp[i]$: The min hopes required from i^{th} stone.

$dp[0] = 1 + \min(dp[1], dp[2], dp[3] \,..\, \boxed{dp[n]})$

rule: pick the farthest one.

$\boxed{dp[n] = 0}$

Dynamic Programming



i

$a[i]$

permute the array as you wish.

$$S = \sum_{1 \le i \le n} a[i]^{i}$$

Maximize

$$dp[S] = \max_{1 \le i \le n} \left(a[i]^{n} + dp[S \setminus \{a[i]\}] \right)$$

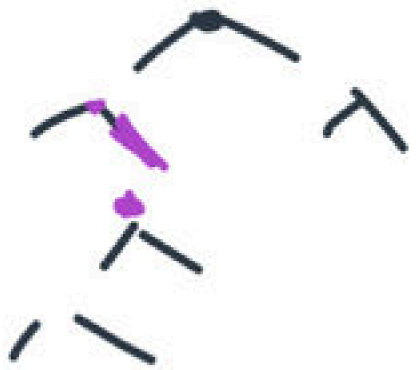
max amongst the array elements.

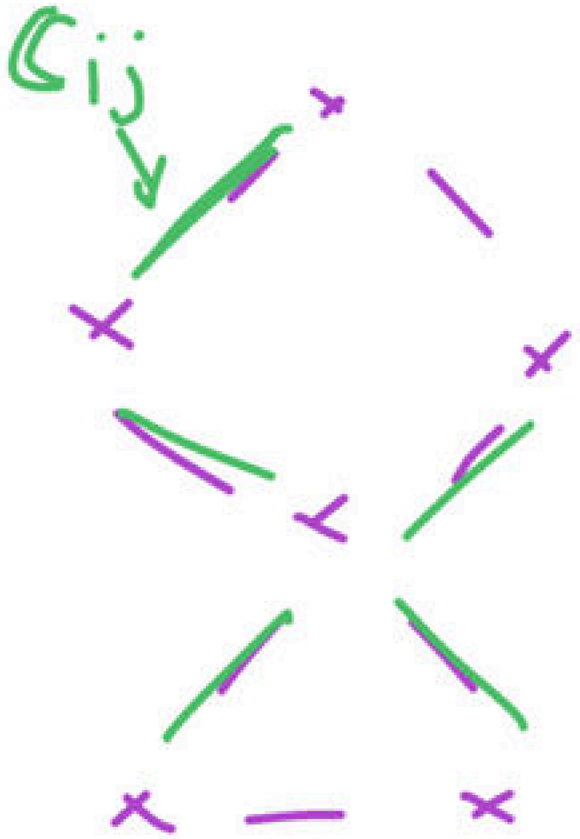
$\#\text{states} = 2^{n}$

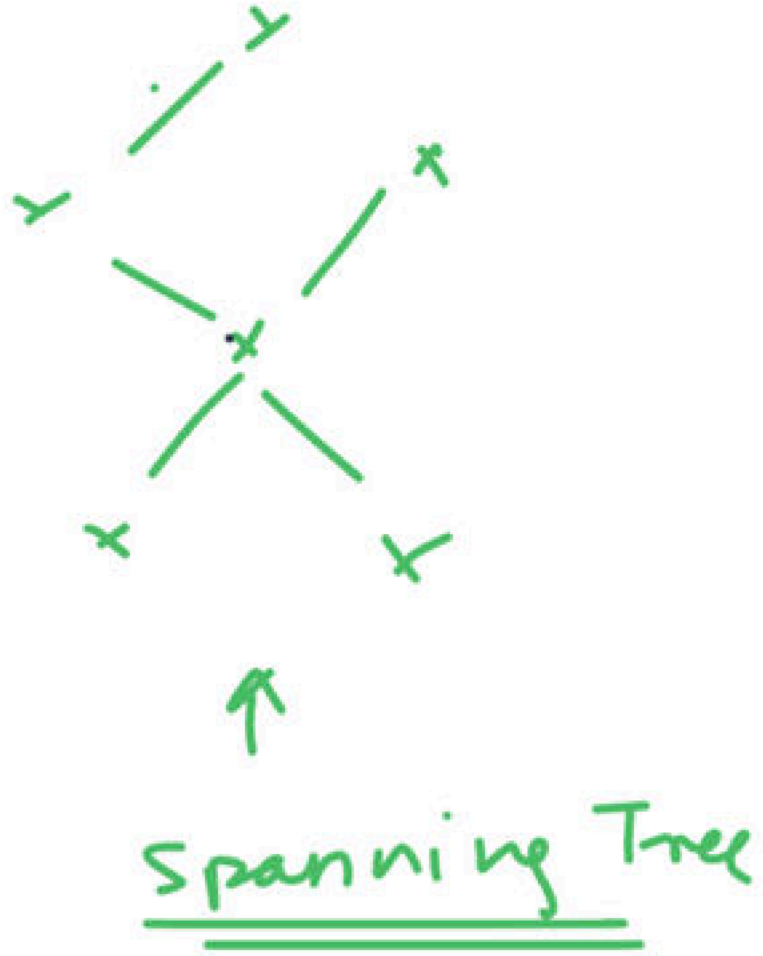
$T.C. = O(n \cdot 2^{n})$

rule: pick the maximum guy.

Minimum Spanning Tree



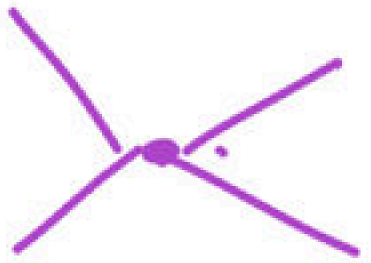




$$dp(G) = \min_{e \in L[u]} \left(cost[e] + dp(G') \right)$$

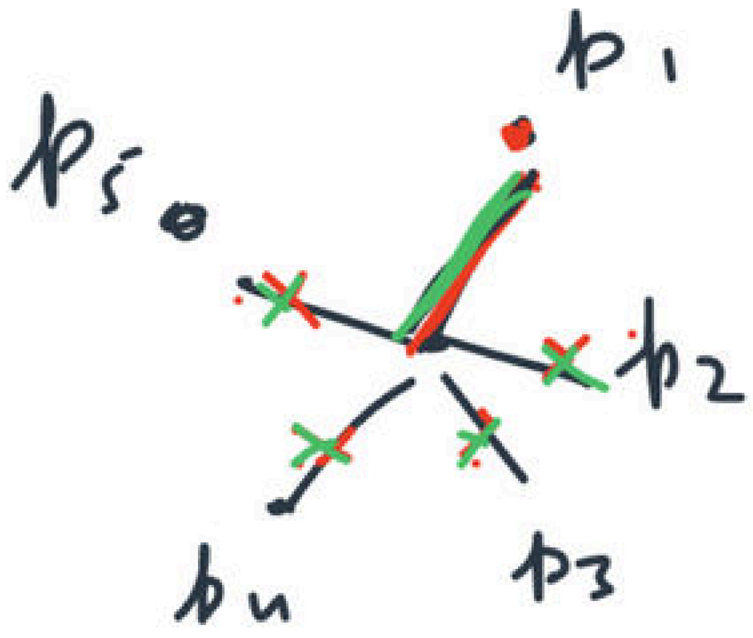
Q Is Greedy and DP inter-related?? → Yes

* You're to min / maximize something. → optimization problem.



Iterate through everything and basically pick the best amongst them.

↳ They are complex or T.C. is higher.



"rule"

1) Greedy stays Ahead.

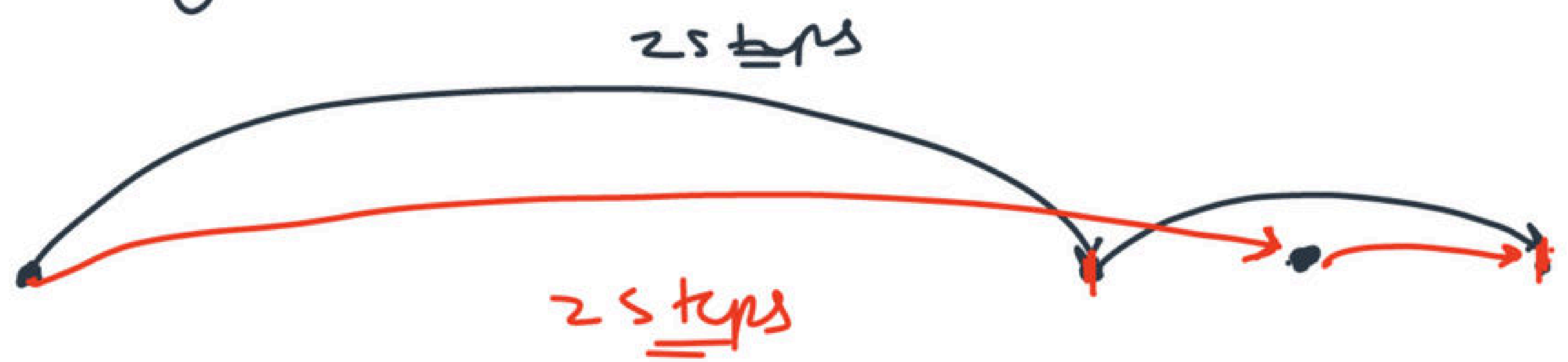
2) Exchange Argument

* You try to optimize locally. $\longrightarrow$ You set some greedy rule.

Q frog Jump

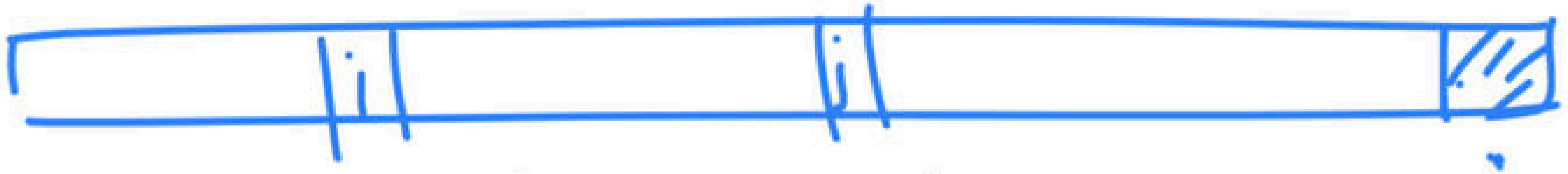
Sol: You jump as far as possible - always.

Assume the contrary that there exists optimal solution where a frog has intentionally not took the largest possible jump.



1) My solution is not inferior
2) If in your optimal solution, there are some small jumps then I can modify them one at a time to arrive at my solution without affecting the optimal answer.

Q)



'sort the array'

Let us assume that there exist another optimal solution where the array is not sorted

$$\frac{i < j}{a[i] > a[j]} \rightarrow \underline{a[i]^{i} + a[j]^{j}} \overset{?}{<} \underbrace{a[j]^{i} + a[i]^{j}}_{\text{They were relatively sorted.}}$$

1) So see my solution is better the one that has been provided by you.
$\therefore$ Your sol^n cannot be optimal.

2) You can swap these elements to arrive at a better sol^n. → You've reduced an inversion.

See you at 4:05 PM

$x < y$

$p > q$

$x^p + y^q < x^q + y^p$

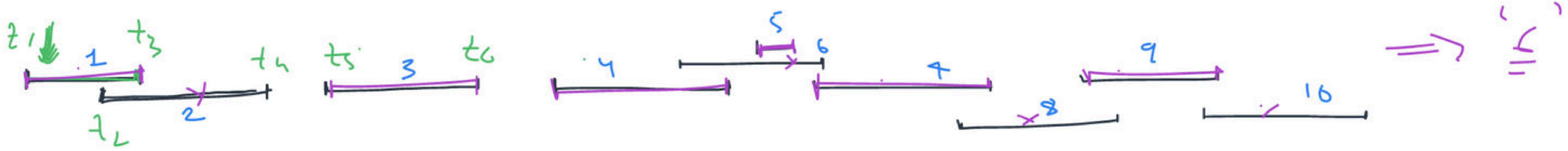
$x^p - x^q < y^q - y^p$

$f(x) < f(y)$

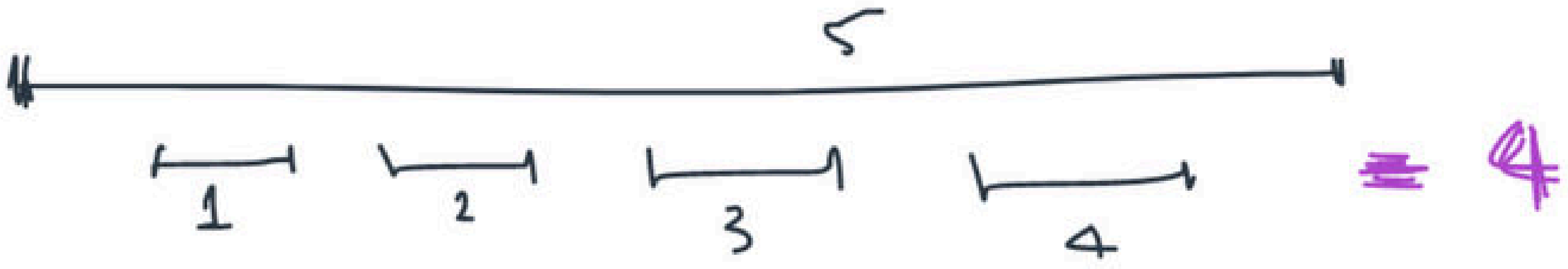
$f(x) = x^p - x^q$ || $f'(x) > 0$ at x

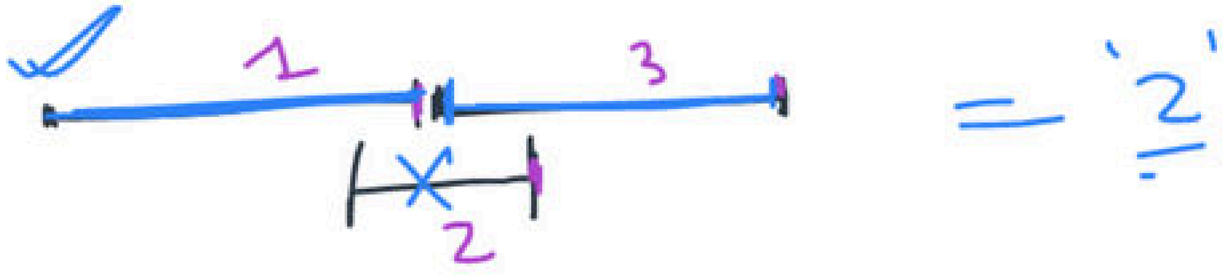
$p x^{p-1} - q x^{q-1} > 0$ $\uparrow$

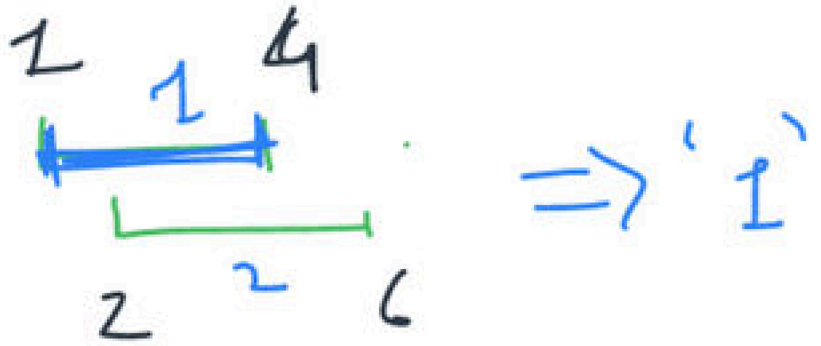
Activity Selection Problem

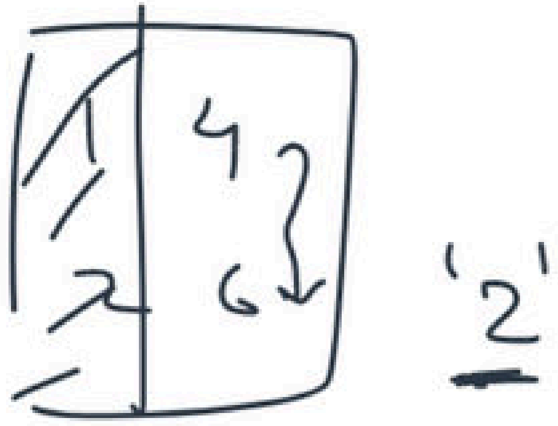


$\boxed{s_i \quad e_i}$ How many max activities can you do in this whole day??

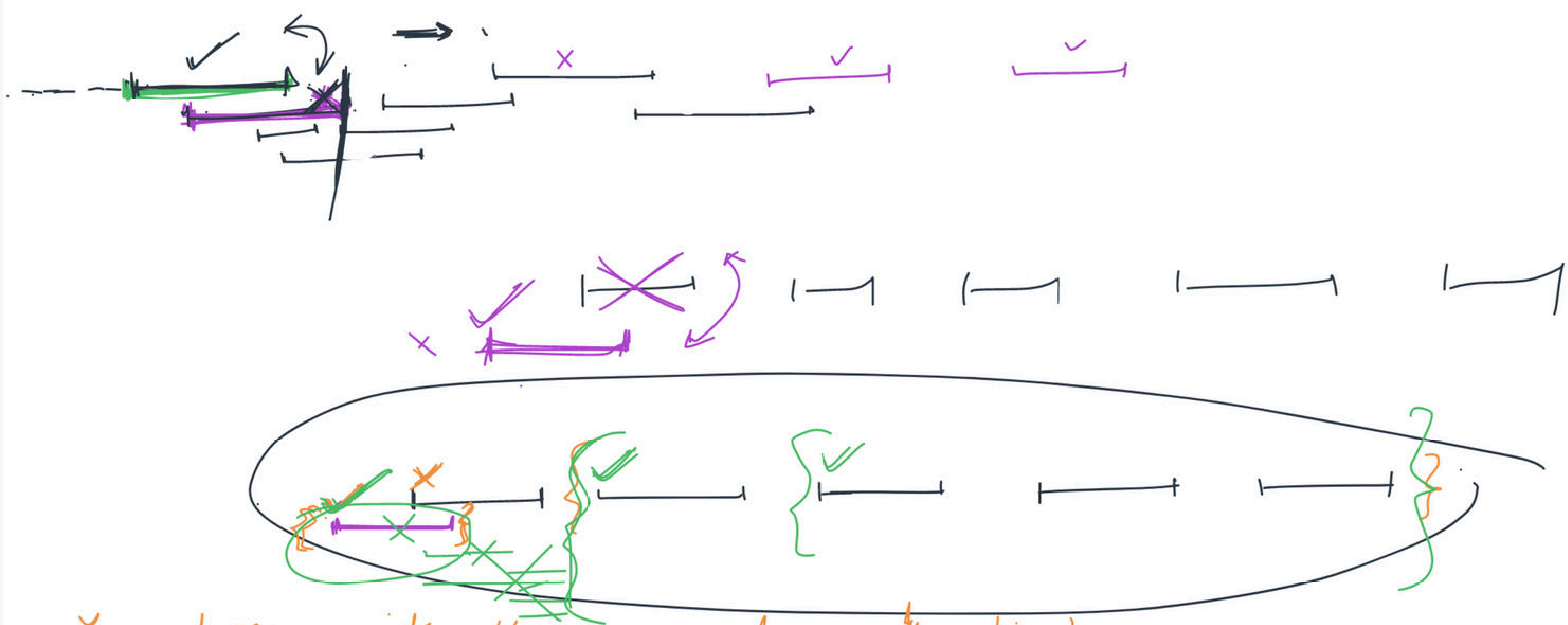








Ans: Sort according to end time and proceed from L → R.



You always pick the guy who ends first.

(8)

[9, 1, 95, 17, 5]

↳ 1) permute in any order

2) concatenate them to produce a big number

largest possible

[1 9, 1 7, 9 5 5] → Aim: Maximize that number

lexicographically

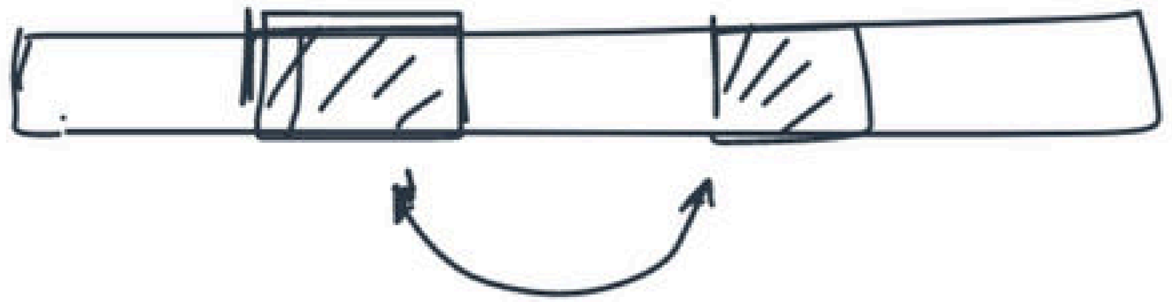
Hypo: Sort according to decreasing order of the numbers

[111, 2] → 1112

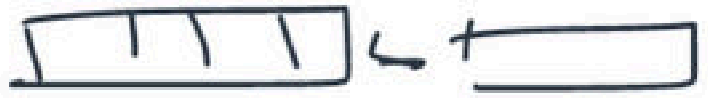
↳ 12111 ?

Assume: Some alternate sol^n is given.

a < b



a > b



1ab

Game Theory and Greedy Algorithms for Interview Preparation

Greedy Algorithms

correct

All possible greedy algorithms, at each step, choose what they know is going to lead to an optimal solution for the general problem

A. True ✓

B. False

All possible greedy algorithms, at each step, choose what they know is going to lead to an optimal solution for the general problem

A. True
B. False

Which of the following are not a step in designing a greedy algorithm?

A. Cast the problem into 2 or more subproblems for which we make the best greedy choices at subsequent steps.

B. Prove that the greedy choice in each step is the only choice that leads to the optimal solution for the general problem.

C. Neither are steps in designing a greedy algorithm

D. They are both steps in designing a greedy algorithm

Which of the following are not a step in designing a greedy algorithm?

A. Cast the problem into 2 or more subproblems for which we make the best greedy choices at subsequent steps.

B. Prove that the greedy choice in each step is the only choice that leads to the optimal solution for the general problem.

C. Neither are steps in designing a greedy algorithm

D. They are both steps in designing a greedy algorithm

Minimum Number of Jumps required for the frog to cross the pond?

A. 4
B. 5
C. 6
D. Not Possible

```
n = 3 (number of lily pads)
k = 5 (frog's max jump distance)
D = [2,4,3,10]
```

Minimum Number of Jumps required for the frog to cross the pond?

A. 4
B. 5
C. 6
D. Not Possible

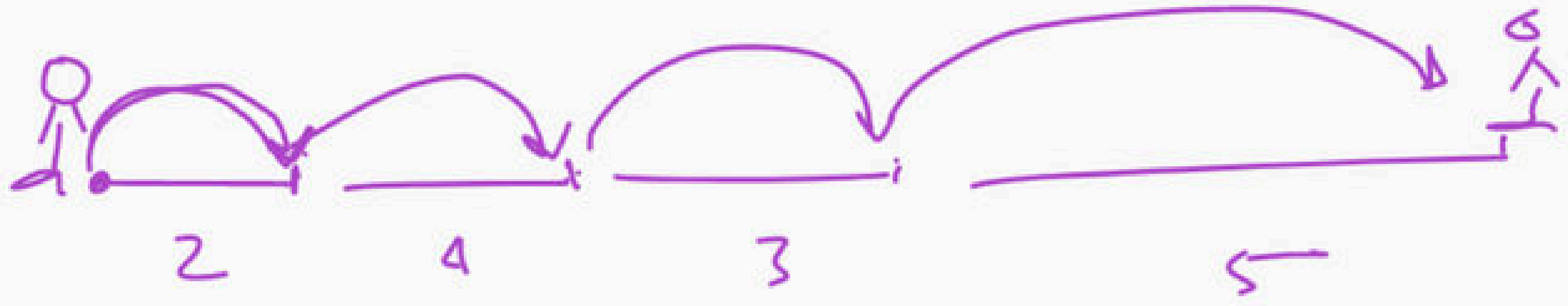
```
n = 3 (number of lily pads)
k = 5 (frog's max jump distance)
D = [2,4,3,10]
```

7

Minimum Number of Jumps required for the frog to cross the pond?

A. 4
B. 5
C. 6
D. Not Possible

```
n = 3 (number of lily pads)
k = 5 (frog's max jump distance)
D = [2,4,3,5]
```



Minimum Number of Jumps required for the frog to cross the pond?

A. 4
B. 5
C. 6
D. Not Possible

```
n = 3 (number of lily pads)
k = 5 (frog's max jump distance)
D = [2,4,3,5]
```

Which permutation of {5,3,2,1,4} would yield the maximum value of a[i]^i. (1- based indexing)

A. {5,3,2,1,4}
B. {1,2,3,4,5}
C. {5,4,3,2,1}
D. {5,3,4,1,2}

Which permutation of {5,3,2,1,4} would yield the maximum value of a[i]^i. (1- based indexing)

A. {5,3,2,1,4}

B. {1,2,3,4,5}

C. {5,4,3,2,1}

D. {5,3,4,1,2}

Minimum possible number of days?

A. 3
B. 2
C. 4
D. 1

```
n = 3
ai = 5, bi = 2
ai = 3, bi = 1
ai = 4, bi = 2
```

Minimum possible number of days?

A. 3
B. 2
C. 4
D. 1

```
n = 3
ai = 5, bi = 2
ai = 3, bi = 1
ai = 4, bi = 2
```

Minimum possible number of days?

A. 3
B. 4
C. 6
D. 5

```
n = 3
ai = 6, bi = 1
ai = 5, bi = 2
ai = 4, bi = 3
```

Minimum possible number of days?

A. 3
B. 4
C. 6
D. 5

```
n = 3
ai = 6, bi = 1
ai = 5, bi = 2
ai = 4, bi = 3
```

Consider the following 6 activities. Maximum number of activities that can be executed?

A. 3
B. 4
C. 6
D. 5

```
start[]  =  {1, 3, 0, 5, 8, 5}
finish[] =  {2, 4, 6, 7, 9, 9}
```

Consider the following 6 activities. Maximum number of activities that can be executed?

A. 3

B. 4

C. 6

D. 5

start[] = {1, 3, ~~0~~, 5, 8, ~~5~~}

finish[] = {2, 4, 6, 7, 9, 9}

Which of the following is the lexicographically smallest string? S = bba, P = 3

A. aaa
B. aab
C. bba
D. aba

Which of the following is the lexicographically smallest string? S = bba, P = 3

A. aaa
B. aab
C. bba
D. aba

That’s all!