Game on!

Exploring Theoretical Computer Science Through Play

What is “Theory” doing in CS?

Here comes the math!

The Plan

01

The Number Guessing Duel

02

The Guess Blocker Challenge

03

Any math here?

04

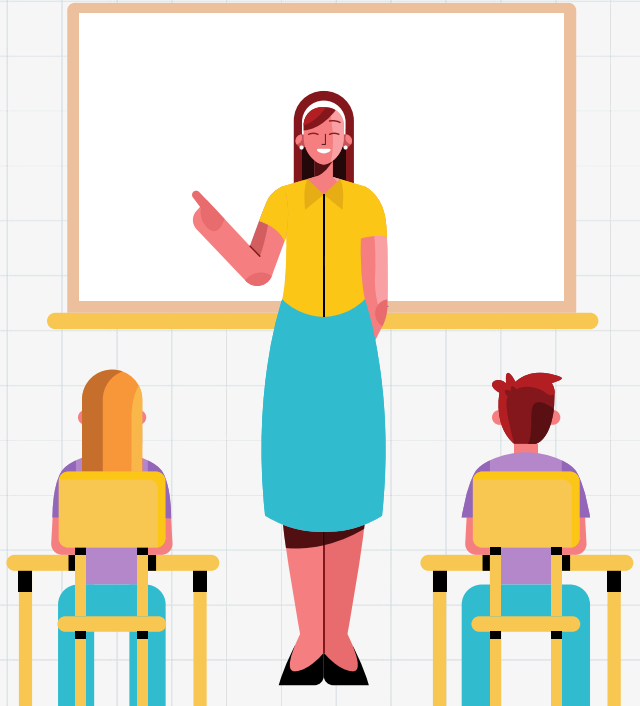
More on TCS!

05

Q&A Session

01

The Number Guessing Duel



Rules of the game

1. It is a 2 player game. The players alternate and play. Numbers 1 to 100 is written on these chits.

2. Each gets to choose a chit. But they do not know beforehand their number.

3. One can ask me any number of questions of the form:

Is the number greater than (lesser than or equal to) x?

where x can be any number.

The first to guess wins!

Is the number

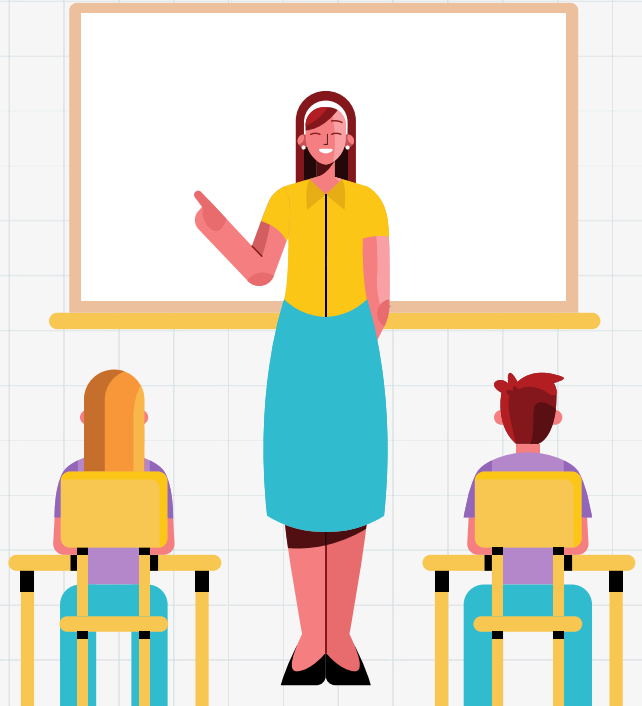
$> x$

Is the number

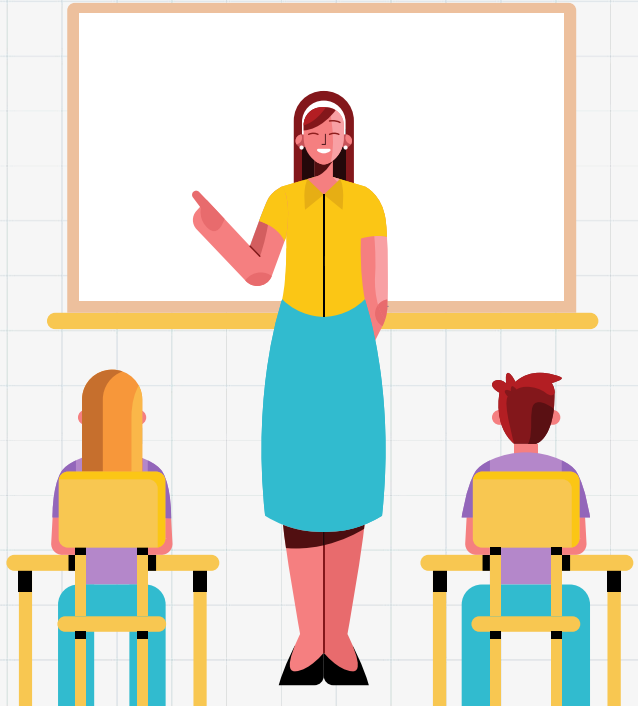
$< x$

Is the number

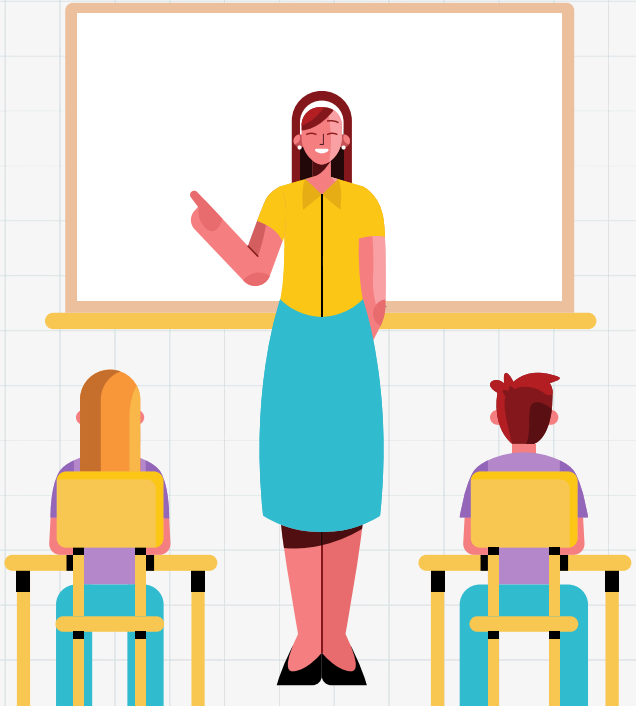
$= x$



Let’s Play!

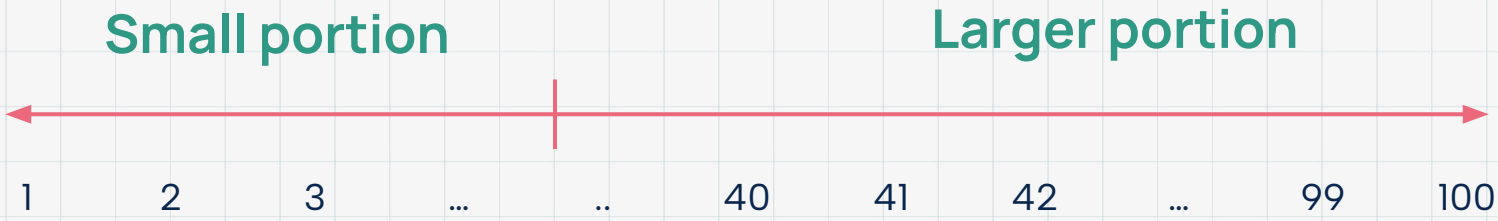


Is there a best strategy?



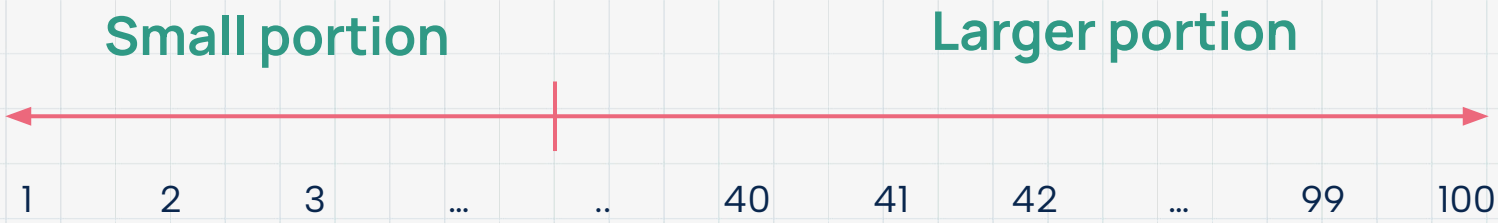
Is there a best strategy?

Try to get rid of a large chunk of possibilities to find the number quickly



Is there a best strategy?

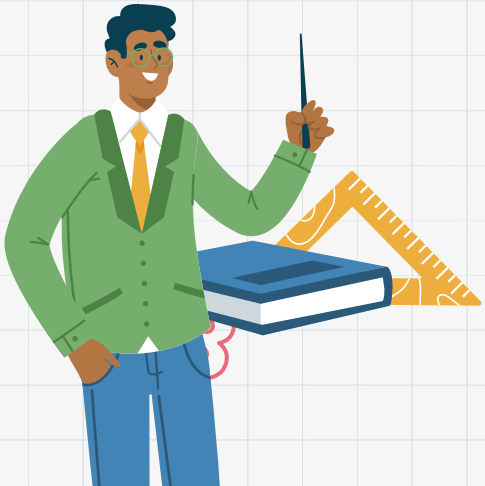
Try to get rid of a large chunk of possibilities to find the number quickly



Is this the best?

02

The Guess Blocker Challenge



Rules of the game

1. Again, it is a 2 player game
2. But now player 1 tries to guess the number player 2 has in her mind
3. The goal of player 1 is to guess the number in least possible number of questions
4. Whereas the goal of player 2 is to prevent this!
5. The player 2 can pick any number that she feels is hard to guess or can come up with any strategy of choosing her number, only condition is that she has to be consistent with all the previously answered questions.

Is the number

$> x$

Is the number

$< x$

Is the number

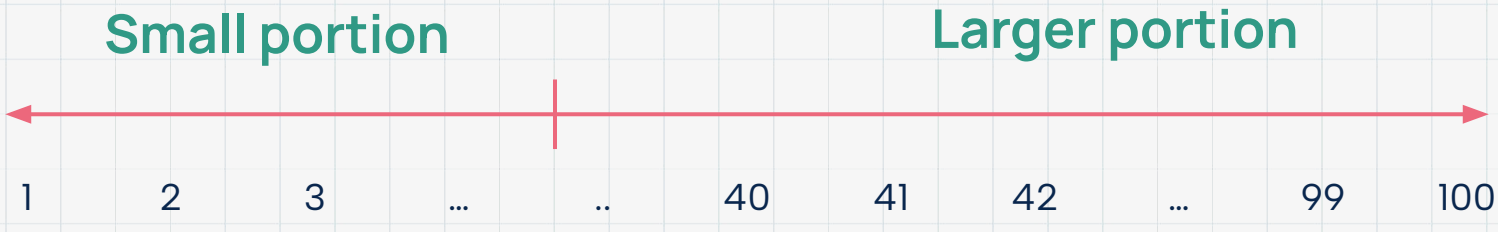
$= x$

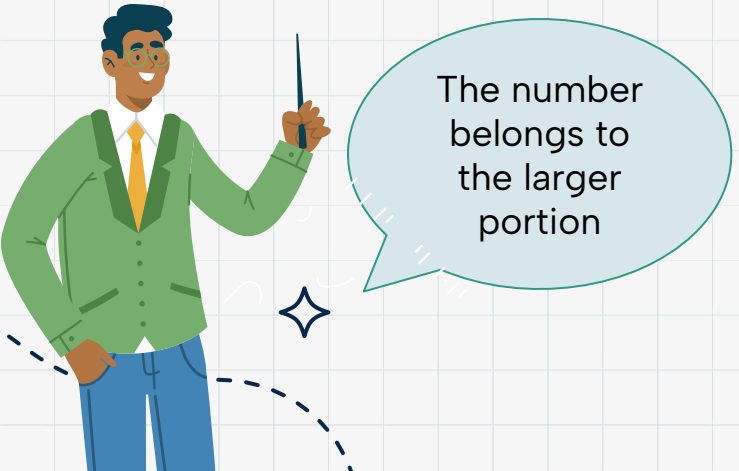
Let’s Play!

Best blocker strategy?

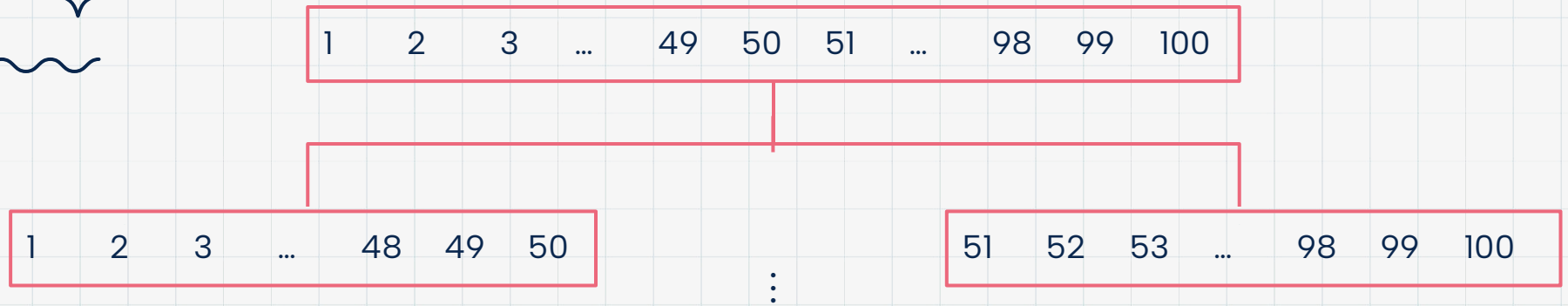
What best a clever blocker can do?

Play against the best strategy of the guesser!

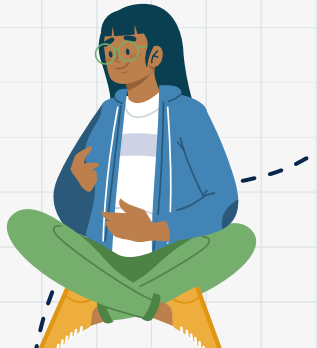




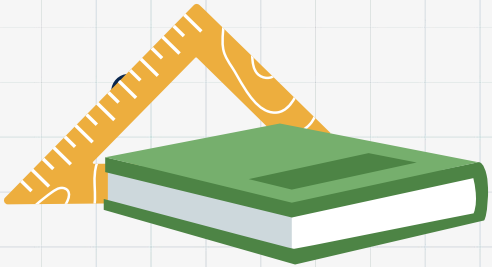
So what best can the guesser do now?



If I try to divide unequally hoping I can get rid of larger chunk, Player 2 can choose not to. So best is to divide equally!



03 Any math here?



Searching in Sorted List of Numbers

- **Our Best Strategy:** Reducing the search space by half every time
- Is this the best one can do?

Can we do better?

Searching in Sorted List of Numbers

- **Our Best Strategy:** Reducing the search space by half every time – **Divide Half**
- Is this the best one can do?

How many *steps* does this take?

steps = ?

Can we do better?

Searching in Sorted List of Numbers

How many *steps* does this take?

steps = ?

$n / 2^{(\#\ steps)} = 1$

$\Rightarrow 2^{(\#\ steps)} = n$

$\Rightarrow$ # steps = $\log_2 n$

It takes **$\log_2 n$** steps to solve using our **Divide Half** strategy

Can we do better?

Is the number

$> x$

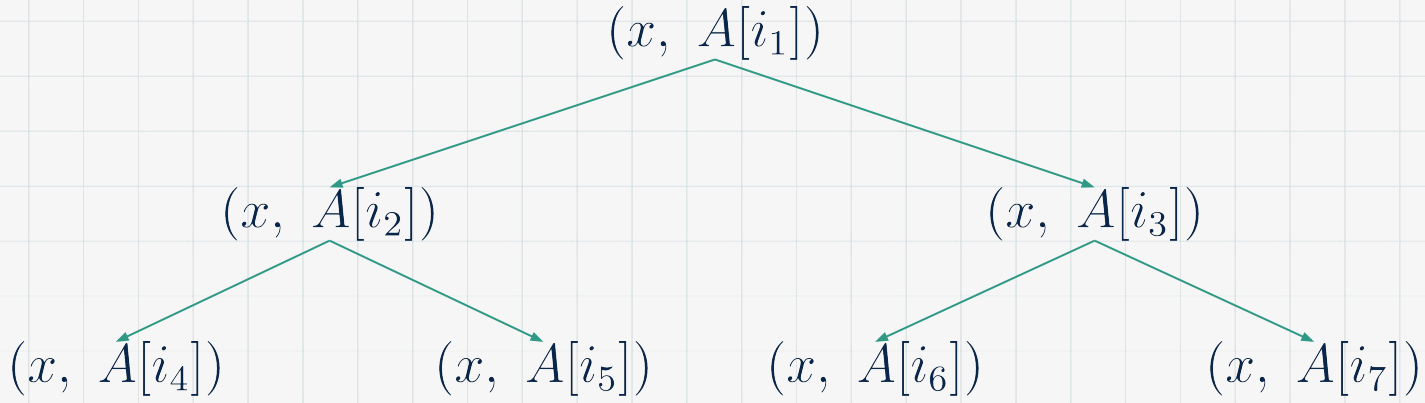
Is the number

$< x$

Is the number

$= x$

Is this the best?

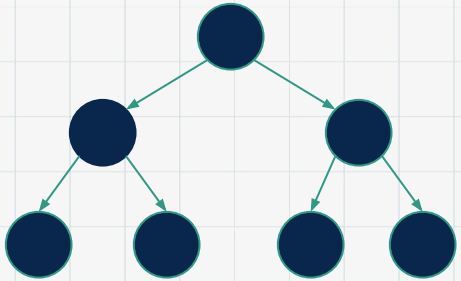


How many leaves does this tree have?

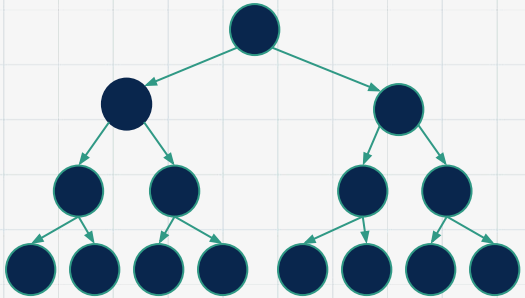
What will be the height of the tree?

How many leaves does this tree have?
n leaves

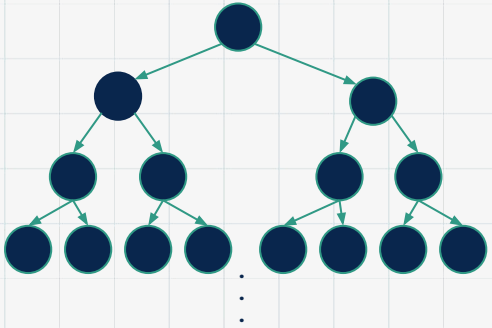
What is the height of the tree?
(Number of steps needed in this process)



Leaves: $4 = 2^2$
Height: $2 = \log_2 4$

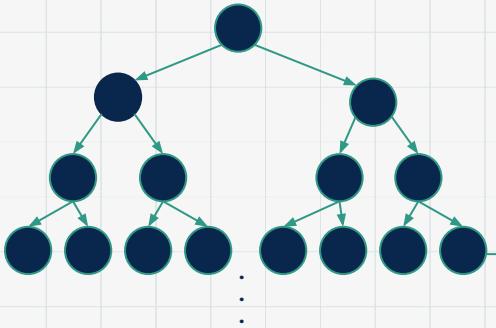


Leaves: $8 = 2^3$
Height: $3 = \log_2 8$



Leaves: $n = 2^{\log n}$
Height: $\log_2 n$

Recall:



Leaves: $n = 2^{\log n}$
Height: $\log_2 n$

 # steps = $\log_2 n$

It takes **$\log_2 n$** steps to solve using our **Divide Half** strategy

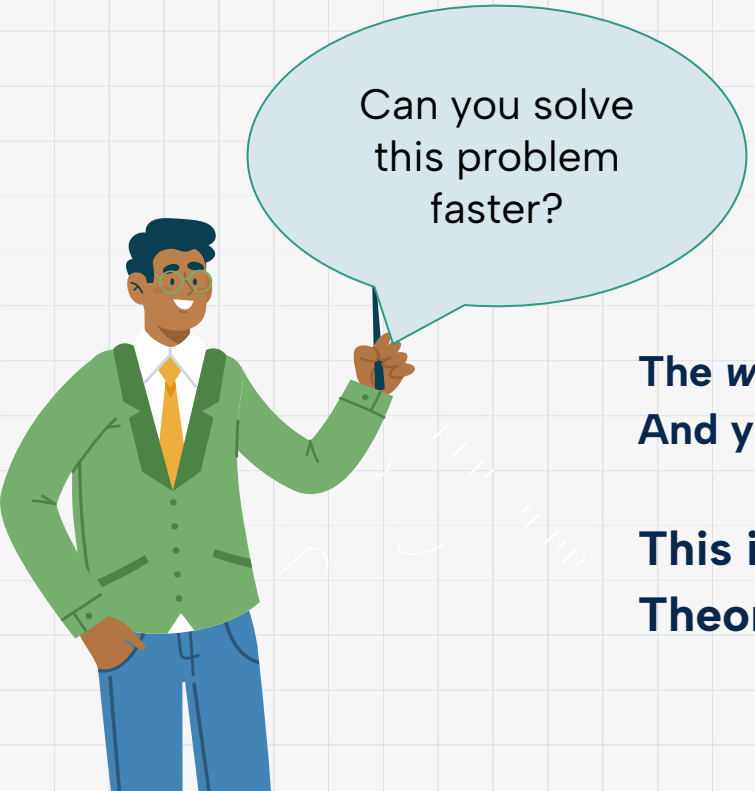
Matches with our best strategy!

What is Theoretical Computer Science?

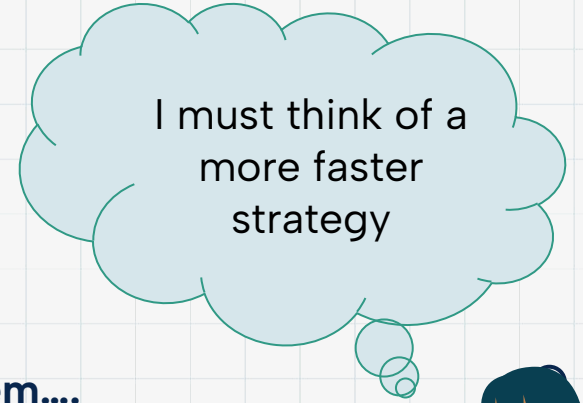
Did we do any TCS now?

Algorithmic thinking!

Art of *efficiently* problem solving

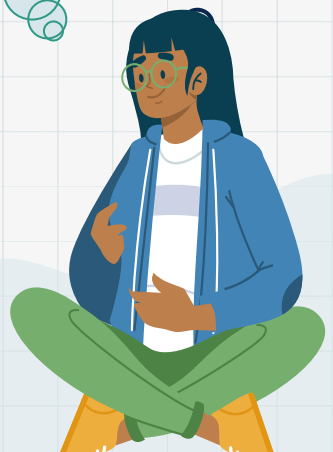


When to stop?

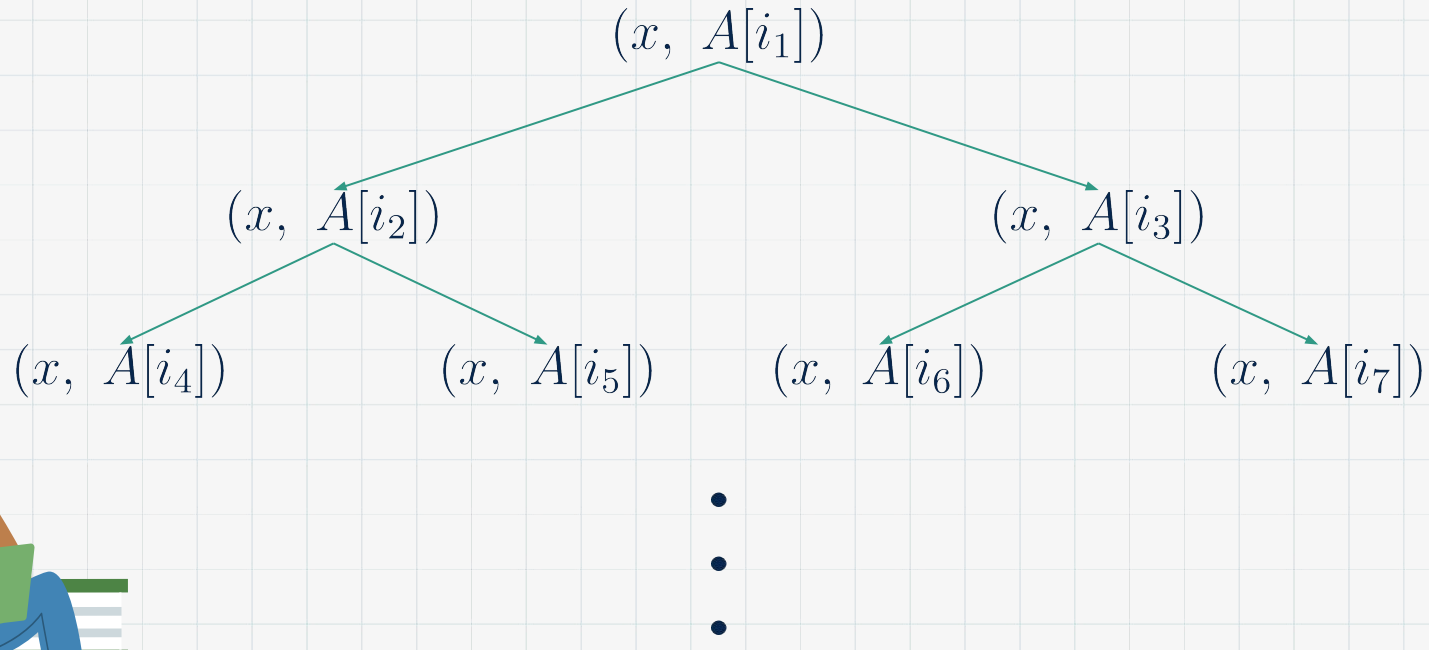


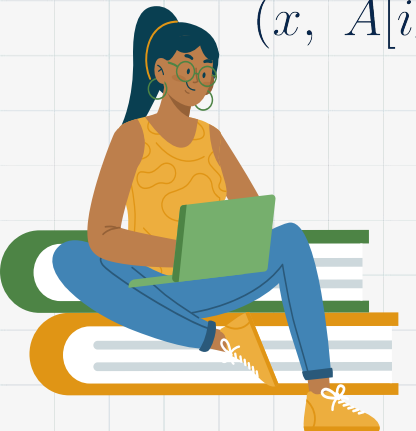
The *world* throws you a problem....
And you wish to find its solution *fast*...

This is the game we play in Theoretical Computer Science



Decision Tree Method





Adversary Method

Not only gives the best possible number of steps solve....

Also gives the best strategy to solve the problem!

04

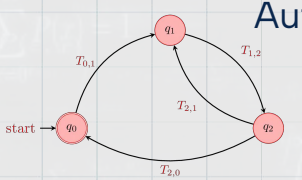
More on TCS!

Theoretical Computer Science

Mathematical logic

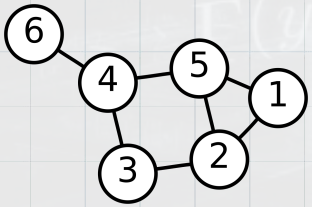
$$P \implies Q$$

Automata Theory

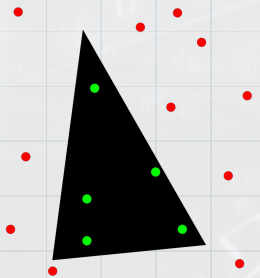


Combinatorics

Graph Theory

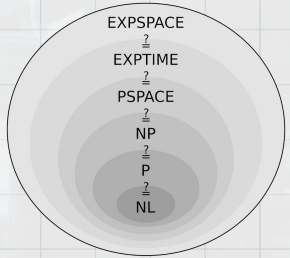


Computational Geometry



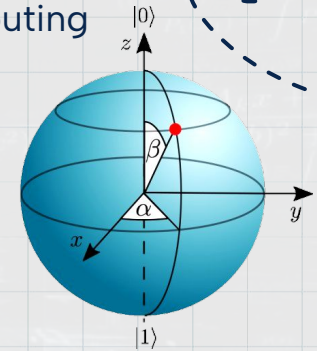
Combinatorial Optimization

Algorithms and Complexity Theory



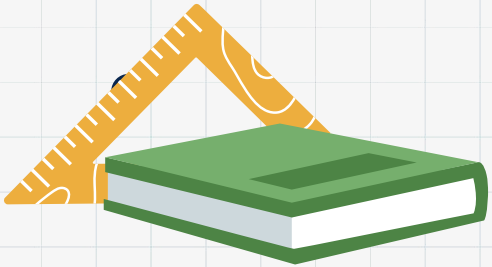
Cryptography

Quantum Computing



And more...

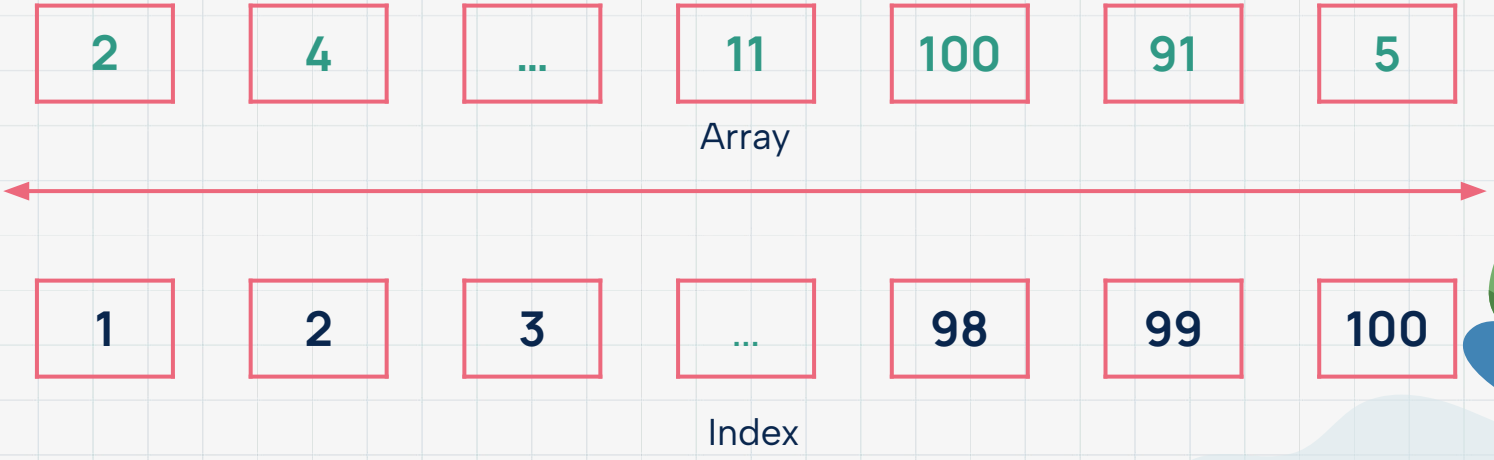
4.5 More Puzzles



More Puzzles

Exercise 1: Prove algebraically that the height of a balanced binary tree is $\log_2 n$ and this is the least compared to any other unbalanced binary tree.

Exercise 2: Think of a faster **non-comparison based** sorting algorithm that takes much less time, given all the numbers in the unsorted array are between 1 and 100. (no repetition)



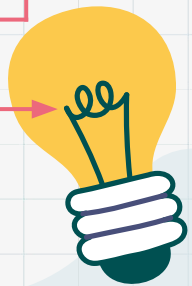
More Using Decision Tree:

Sorting

Exercise 3: Given a list of n distinct elements how many steps do you need to sort the list? At every step you get to compare two elements and sort. Give the best strategy to do this in the least number of steps.

2	4	...	11	100	91	5

Array



More Puzzles

Exercise 4: This exercise introduces you to a new way of asserting that for a given problem a strategy is best, based on an already known best strategy for another problem.

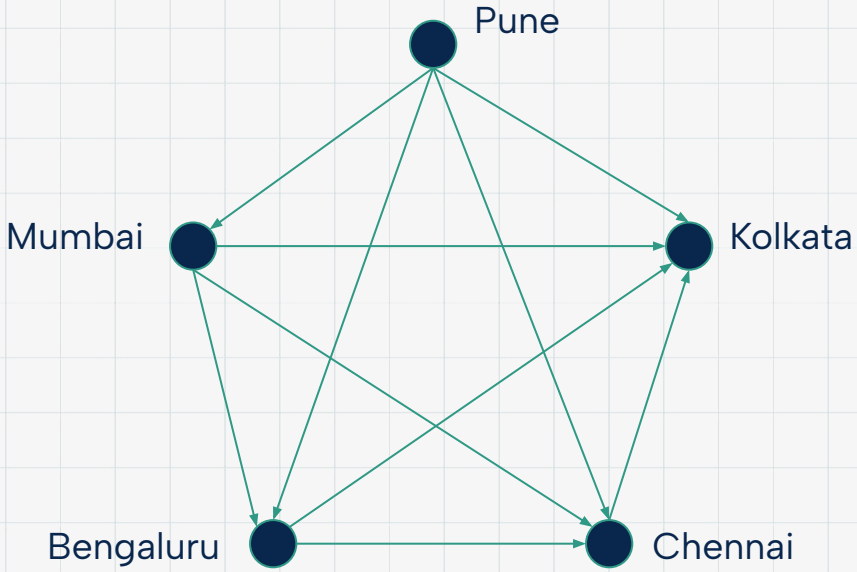
Traveling Salesman Problem

Traveling Salesman Problem*

A salesman wants to travel to 5 cities and there is only one way path directly between two city. He wants to take the shortest route possible. How will you find the shortest route covering all the cities? How many steps does your algorithm take? Can you prove that this is the best you can do?

Note*: This is not the most general way of stating traveling salesman problem(TSP). In specific we are looking at TSP on tournament graph.

Traveling Salesman Problem



Note that the arrows represent the direction. If A –> B this means we can go from A to B, but not directly from B to A. Also note that the task is to find a directed cycle in the above *graph*.

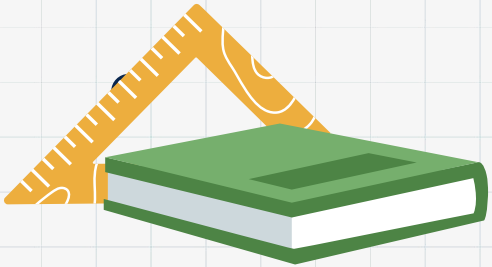
References

- Lecture on Lower Bounds by Prof. Venkatesh Raman, IMSc
- Wikipedia Theoretical Computer Science
- Image credits: Wikipedia images

Nice Read

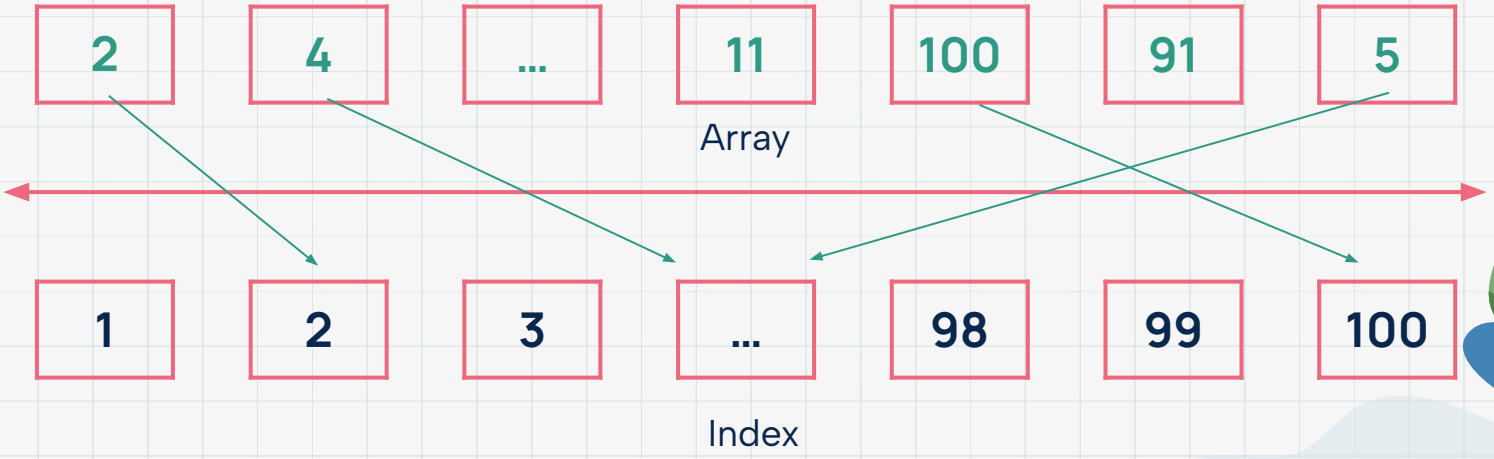
- Algorithms Book by Christos Papadimitriou, Sanjoy Dasgupta, and Umesh Vazirani

4.5 More Puzzle: Solutions



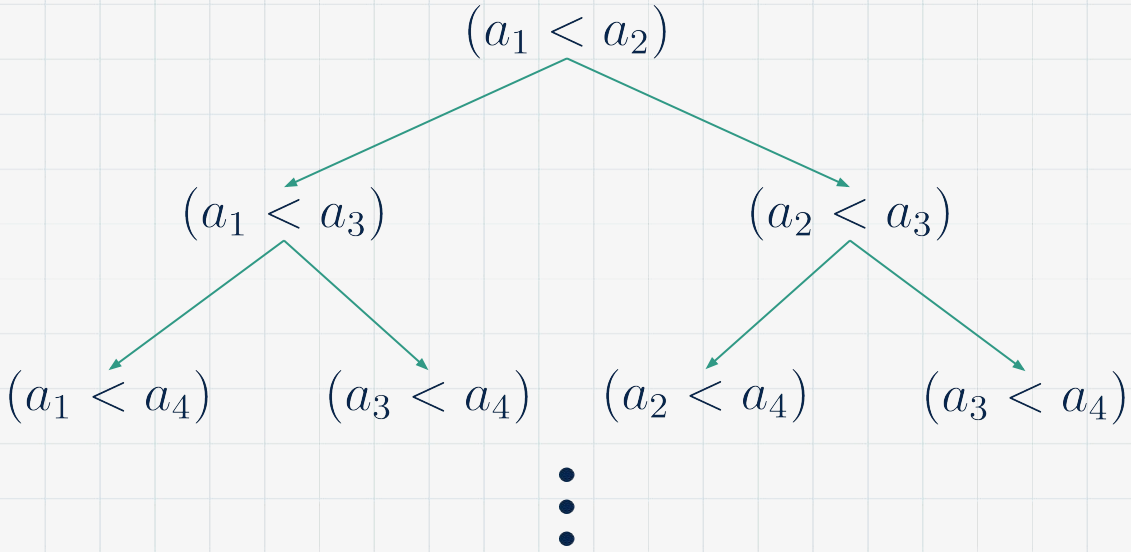
Non-comparison Based Sorting

Have another empty list. Just go over the unsorted list and put every element in the new list where its value matches the index in the new list.



More Using Decision Tree:

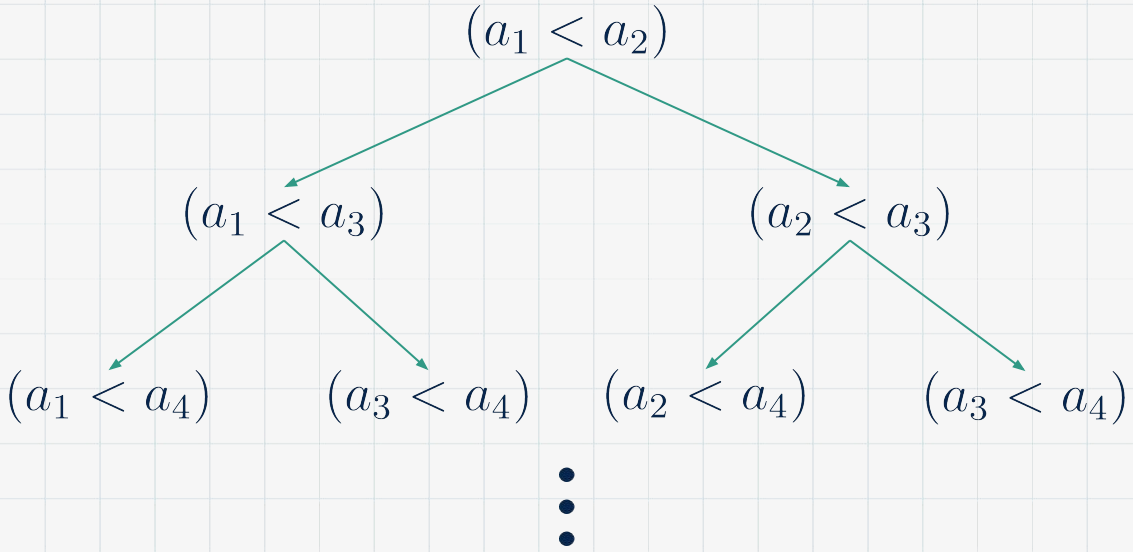
Sorting



How many leaves does this tree have?

More Using Decision Tree:

Sorting

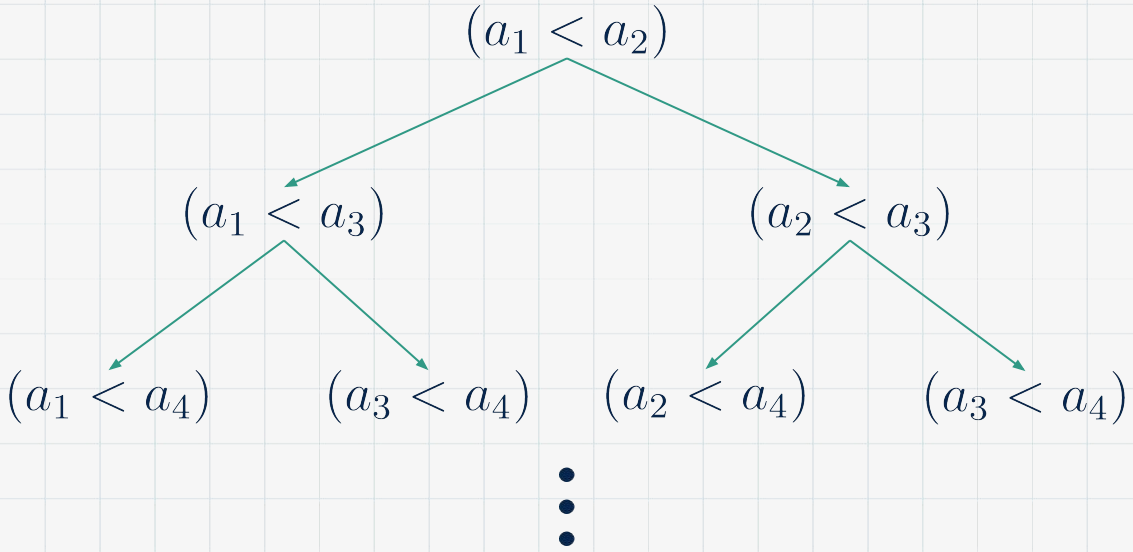


How many leaves does this tree have?

n! many permutations

More Using Decision Tree:

Sorting



Height of the tree?

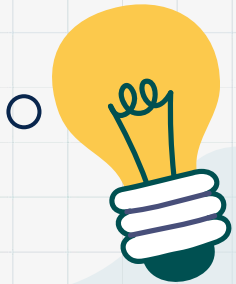
$\log_2 n!$

Fact:

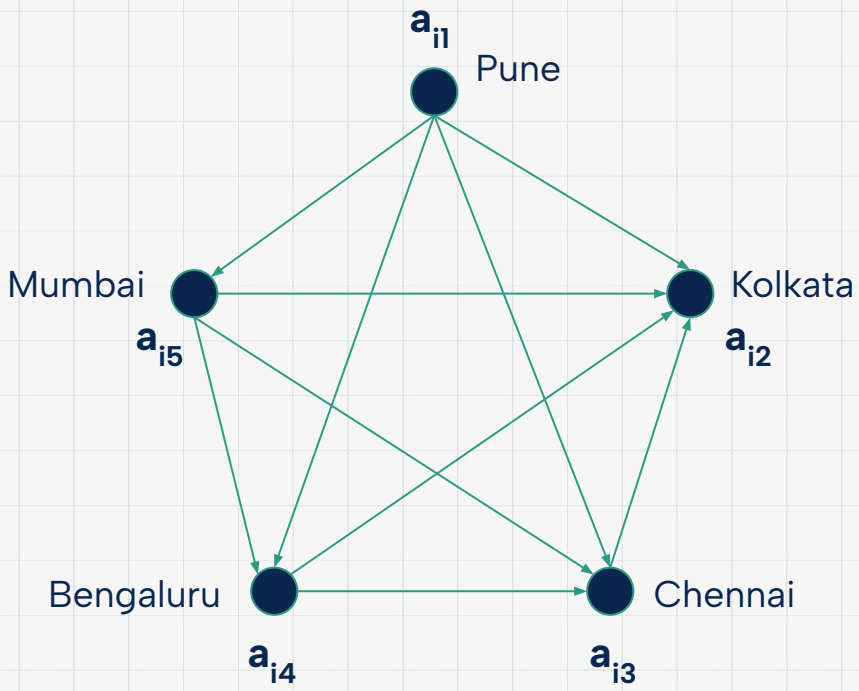
$\log_2 n! \leqq n \log_2 n$

As 1 X 2 X .... n $\leqq$ n X n X .... n
Therefore, $\log_2 n! \leqq \log_2 n^n = n \log_2 n$

Thus *sorting* can not be done faster than **n logn**



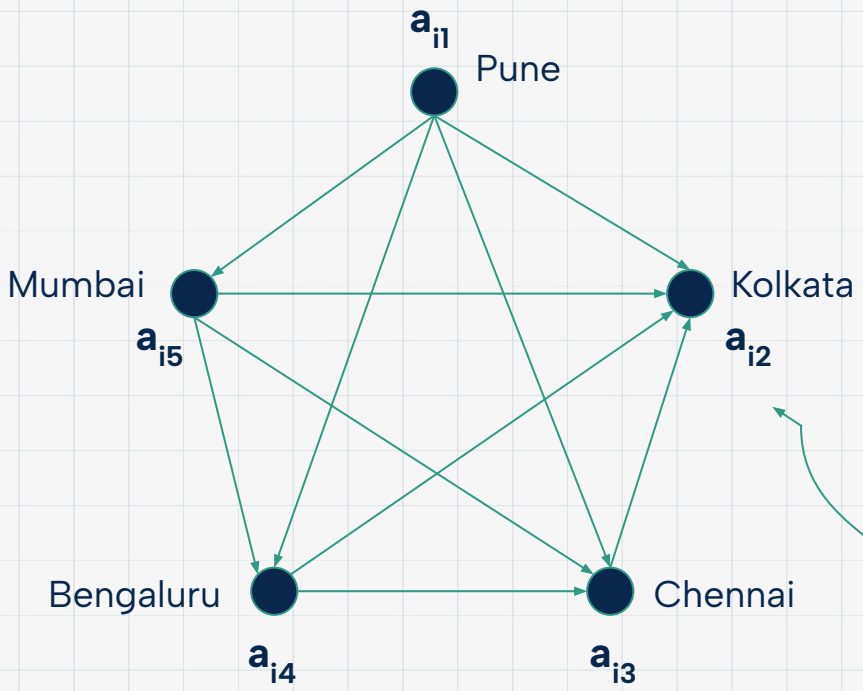
Traveling Salesman Problem



$a_1 = 1$
$a_2 = 2$
$a_3 = 3$
$a_4 = 4$
$a_5 = 5$

Note*: This is not the most general way of stating traveling salesman problem(TSP). In specific we are looking at TSP on tournament graph.

Traveling Salesman Problem

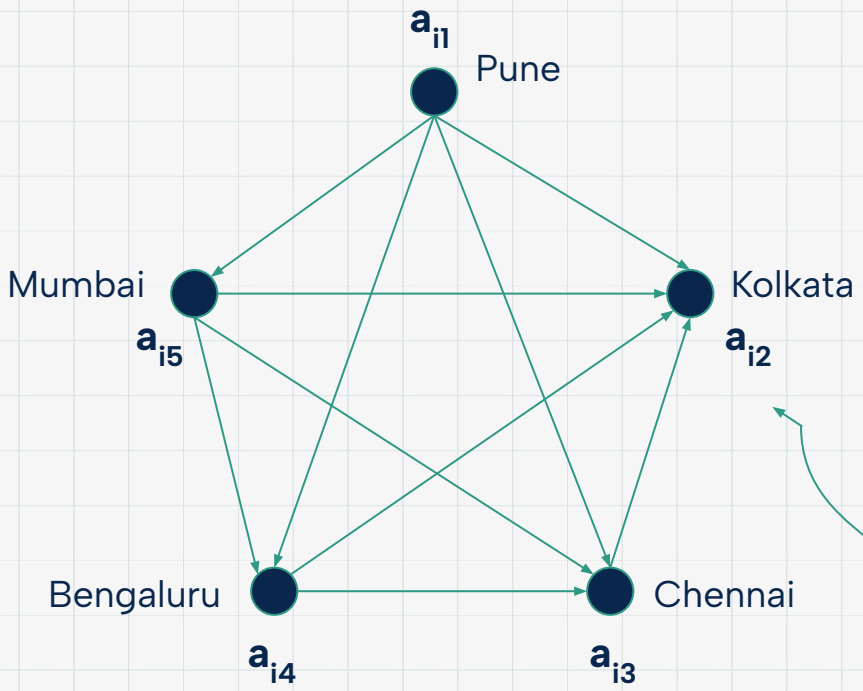


$a_1 = 1$
$a_2 = 2$
$a_3 = 3$
$a_4 = 4$
$a_5 = 5$

Given n items a_1 , a_2... a_n model it as *such* a graph where $a_i < a_j \Leftrightarrow i \rightarrow j$ in the graph. A directed path covering all cities corresponds to sorted order of the elements

Note*: This is not the most general way of stating traveling salesman problem(TSP). In specific we are looking at TSP on tournament graph.

Traveling Salesman Problem



$a_1 = 1$
$a_2 = 2$
$a_3 = 3$
$a_4 = 4$
$a_5 = 5$

Thus we can not do better than **nlogn** for traveling salesman problem*

Given n items a_1 , a_2... a_n model it as *such* a graph where $a_i < a_j \Leftrightarrow i \rightarrow j$ in the graph. A directed path covering all cities corresponds to sorted order of the elements

Note*: This is not the most general way of stating traveling salesman problem(TSP). In specific we are looking at TSP on tournament graph.

Thanks!

More to ask?

quantaoncomputing@gmail.com

https://o-qcblog.github.io/