LVX
VERITAS
VIRTVS

Lesson 1:

Np-Completeness

P, NP and NP-Hard problems

INFO6205: Program Structure and Algorithms

Northeastern University
College of Engineering

Table of Contents:

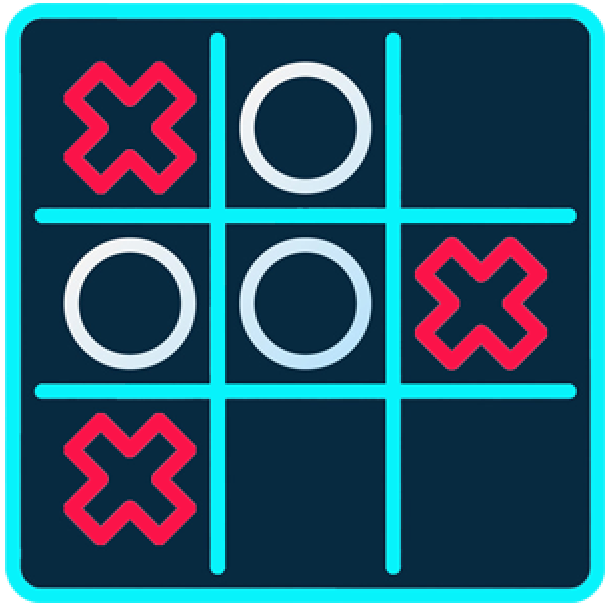
Man vs Machine

https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov

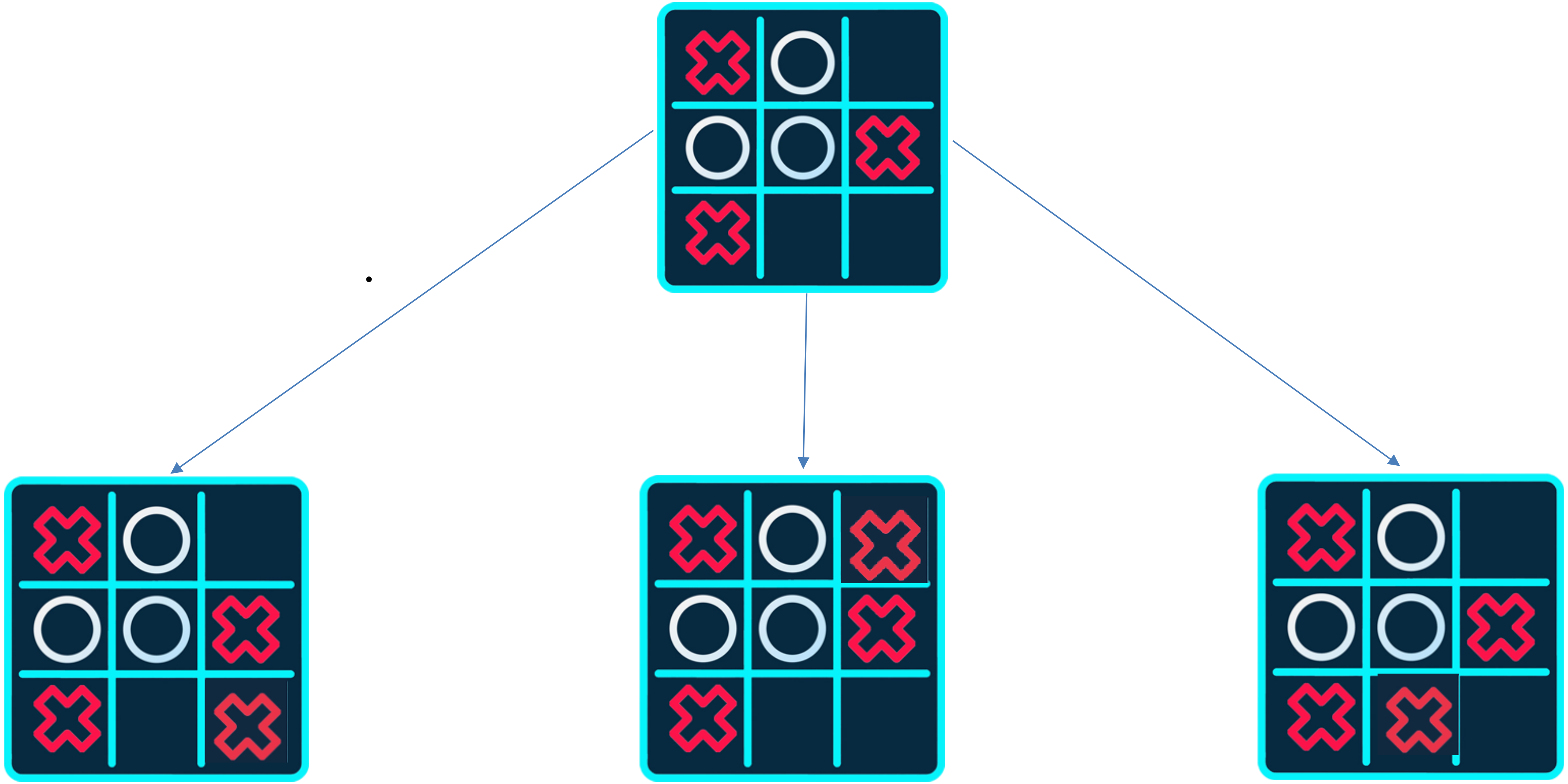
How was Deep blue a computer program able to defeat the world chess champion?

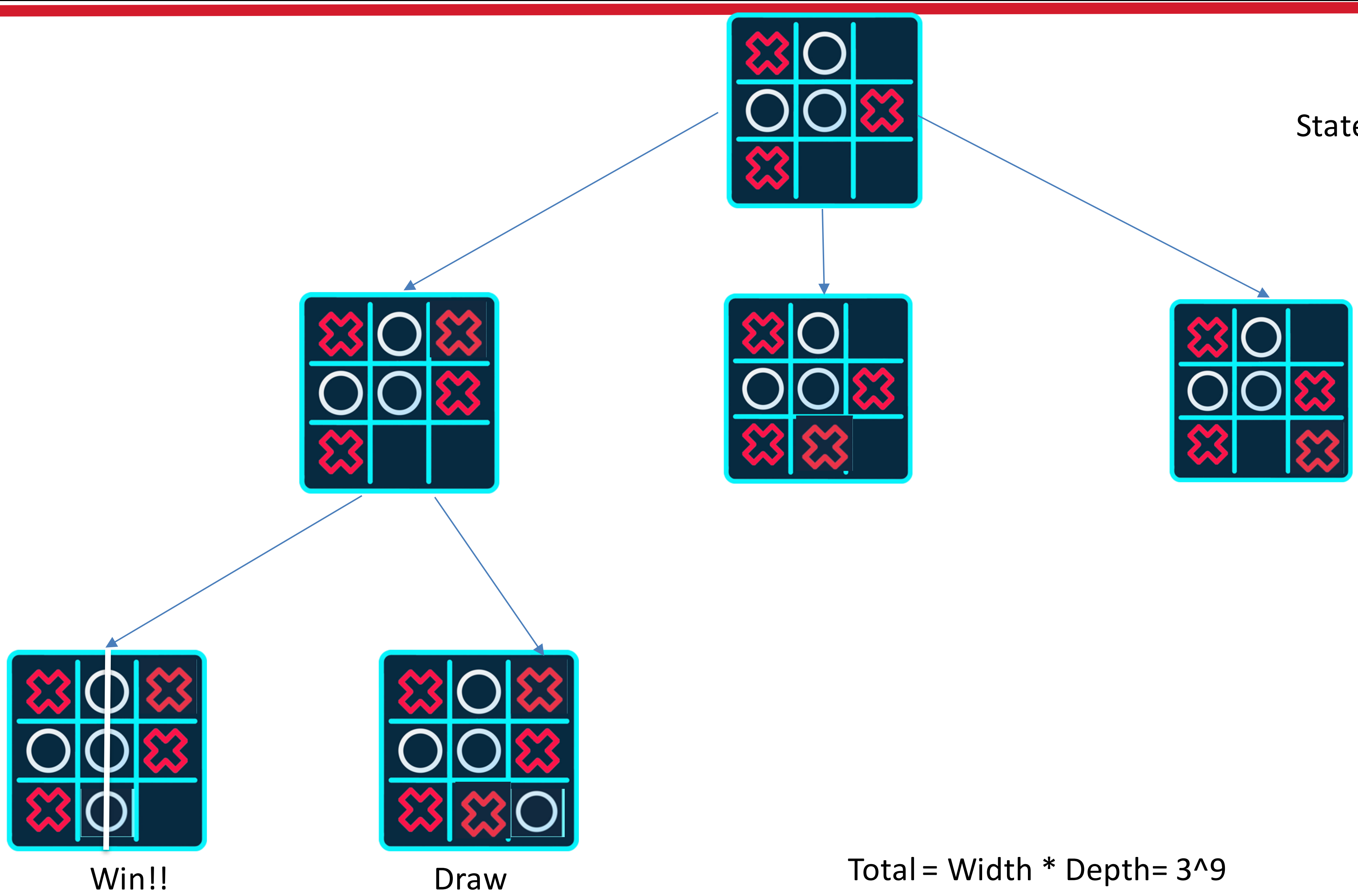
Let's figure it out !!

Let's start with something simple !



If you are X, where would you place it?





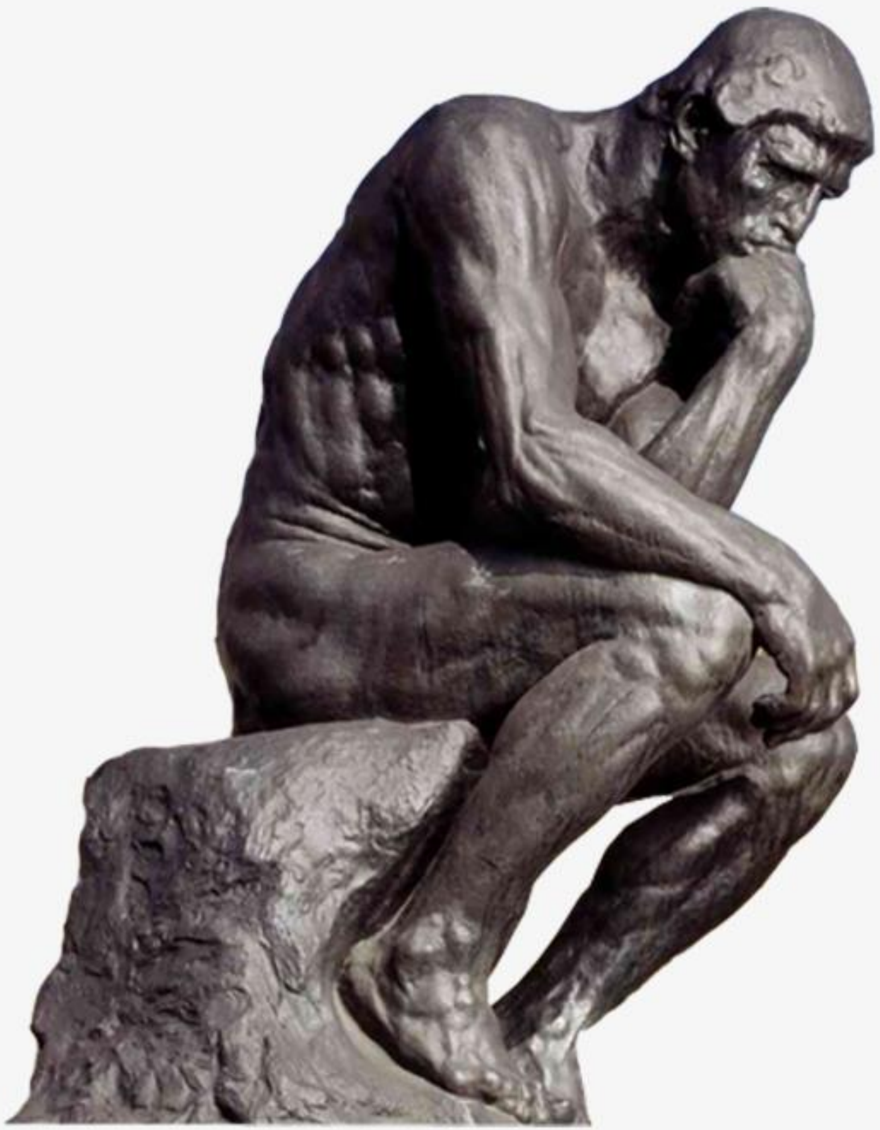
States: 1

3

9

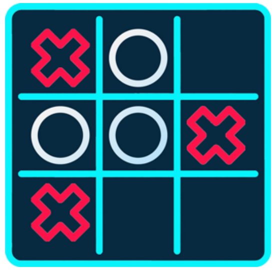
Total = Width * Depth= 3^9

How many states? 19683



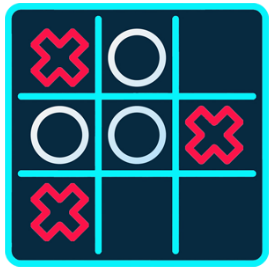
Is it possible to write a program that plays Tic-Tac-Toe?

https://gist.github.com/qianguigui1104/edb3b11b33c78e5894aad7908c773353



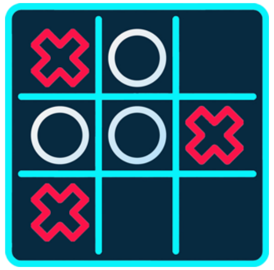
Is the game over?

Constant



Is this game winning for `O`?

60% chance (partial tree)



Win this game for `O`

Compute all states
(complete tree)


What if N is large, do we know any such game?

Has black or white won?

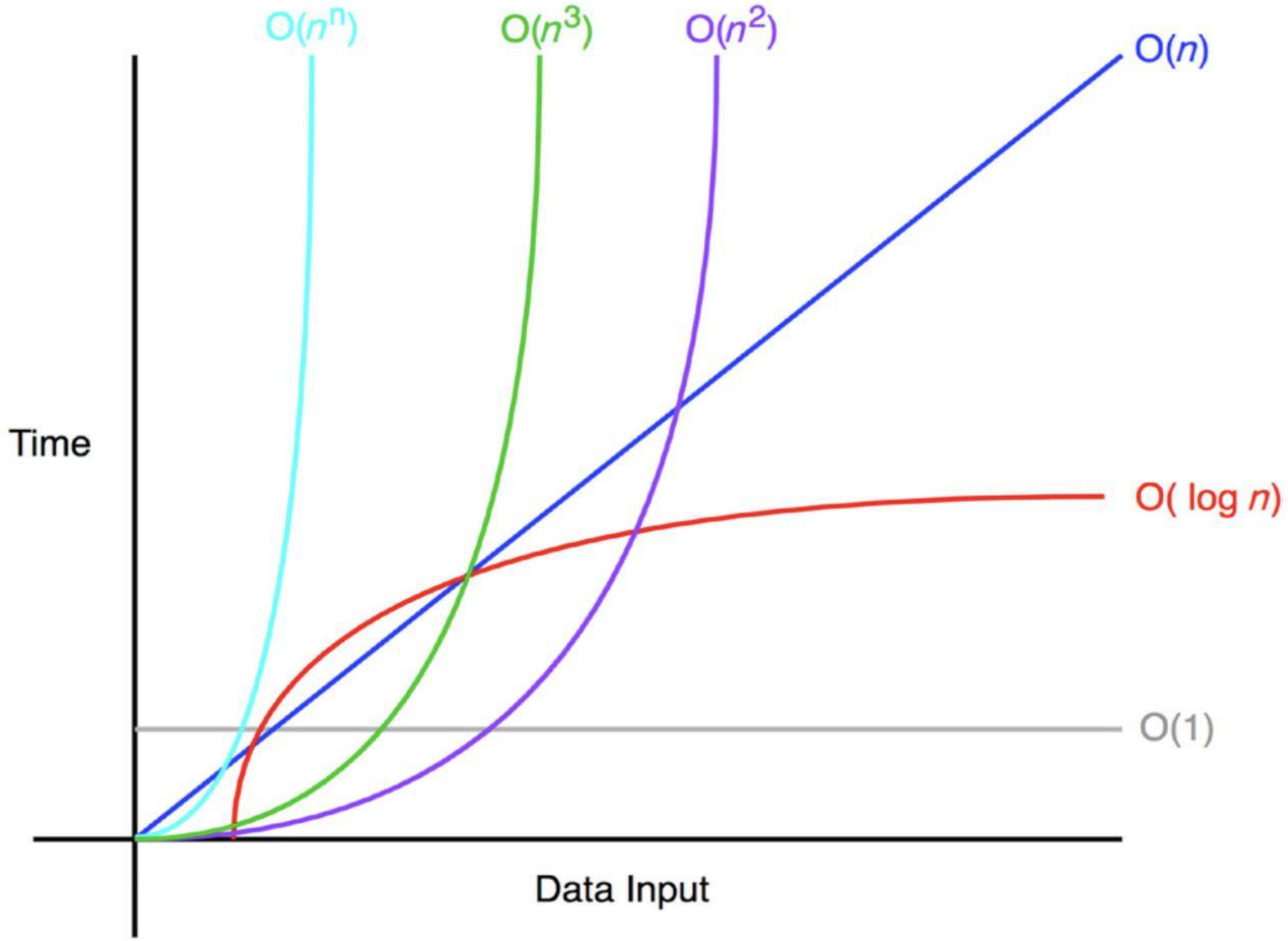
Constant

Given the state, is the game wining for black?

60% chance (partial tree)

Make Black win

Compute all states (Complete Tree)


O(n^n)
O(n^3)
O(n^2)
O(n)
Time
O(log n)
O(1)
Data Input

Implications and Applications

1. **Machine Learning and Data Science**

Many machine learning tasks involve optimization, where algorithms seek optimal solutions based on certain criteria. These criteria can include minimizing errors, maximizing likelihoods, or reducing loss functions.

Understanding the computational complexity of optimization problems helps researchers choose appropriate algorithms for training models efficient.

Feature selection involves choosing the most relevant subset of features from a larger set. This can be framed as NP problem, where the best subset must be found while considering all possible combinations.

The NP-Completeness of such problems influences how researchers approach feature selection, often leading to heuristic-based methods.

2. Cryptography and Security:

Many cryptographic protocols rely on the assumption that certain problems are computationally hard to solve. For example, the security of public-key cryptography is based on the difficulty of factoring large composite numbers.

The NP-Hardness and intractability of problems like Integer Factorization play a central role in establishing the security of encryption schemes.

One-way functions are functions that are easy to compute but hard to invert. These functions are fundamental to many cryptographic applications, including digital signatures and password hashing.
The security of public-key cryptography relies on the assumption that certain problems are difficult to solve, such as the Discrete Logarithm Problem or the RSA Problem.

These problems are believed to be hard even for powerful adversaries equipped with advanced computational resources.

Conclusion

P class: Problems that can be solved efficiently in polynomial time using deterministic algorithms.

NP class: Problems for which a potential solution can be verified in polynomial time.

NP-Complete class: Problems that are among the hardest problems in NP, with the property that any NP problem can be reduced to them in polynomial time.

NP-Hard class: Problems that are at least as hard as the hardest problems in NP, even though they might not be in NP themselves.

The P vs. NP question remains one of the most significant open problems in computer science exploring whether P = NP or P ≠ NP.

Computational complexity concepts have wide-ranging practical applications, from optimization and cryptography to machine learning and data science.

These concepts guide the design of efficient algorithms, inform decisions in various fields, and underpin the security of modern cryptographic systems.

Time For a Quiz!

Question: Why is the task of matching drivers to riders and optimizing routes considered challenging for ride-hailing platforms like Uber and Lyft?

A) Because it involves verifying user information
B) Because it requires efficient driver background checks
C) Because it's similar to NP problems, where solutions are easily verified
D) Because it's a one-time optimization task with simple algorithms

Question: Why is handling surge pricing during peak demand considered a challenging aspect of ride-hailing platforms like Uber and Lyft?

A) Because it involves calculating driver earnings accurately
B) Because it requires real-time monitoring of driver locations
C) Because surge pricing is an NP-complete problem
D) Because it's difficult to verify the legitimacy of surge events