All About

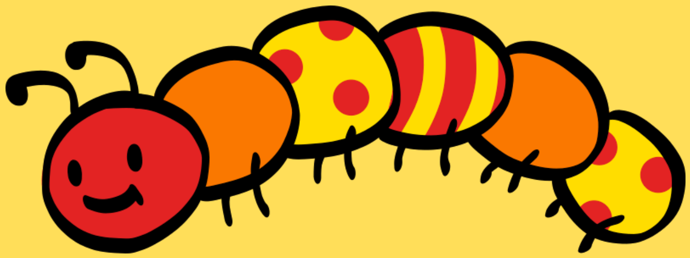
Time & Space Complexity

@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

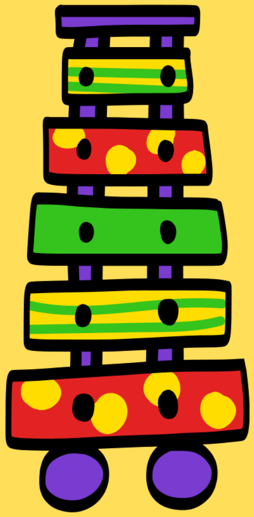
Time Complexity :

- *Time complexity is a measure of how long an algorithm takes to complete its task.*
- ***It's like measuring how much time the algorithm needs to run and give the output, based on the size of the input.***



Example :

- *Let's say you're searching for a name in a phone book and the phone book has 100 pages.*
- *You're searching sequentially through each page until you find the name.*



- *Now here we can face two extreme cases : Best case and Worst case.*

@Maheshpal Singh Rathore
@mpsrathore2020

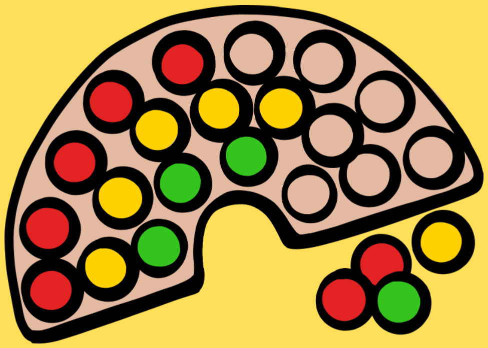
SWIPE >>>

- *In case one ,the best-case scenario is that the name is on the first page.*

- *So you'll have to search through only 1 page.*

- *In case two ,the worst-case scenario is that the name is on the last page.*

- *So you'll have to search through all 100 pages.*



Space Complexity :

- ***Space complexity is a measure of how much memory an algorithm requires to complete its task.***

- ***It's like measuring how much space the algorithm needs to store the input data and perform any necessary operations on that data.***

@Maheshpal Singh Rathore
@mpsrathore2020

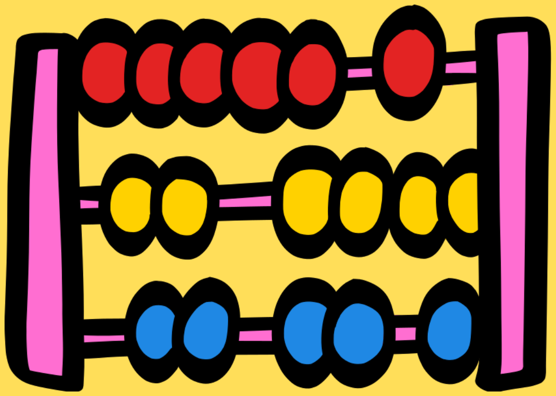
SWIPE >>>

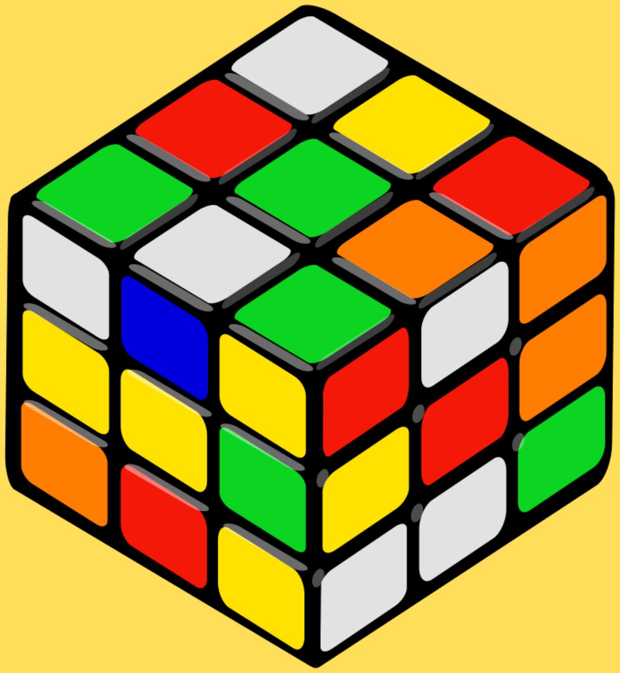
Example :

- **Let's say you have an algorithm that sorts a list of numbers.**

- **To do this, the algorithm might need to create a new list to hold the sorted numbers, in addition to the original list of unsorted numbers.**

- The space complexity of the algorithm would be the amount of memory required to store both lists.





SWIPE >>>

ASYMPTOTIC NOTATION :

- ***Asymptotic notation is the shortest way to represent the Time complexity of an algorithm .***

- ***That is , it is used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value .***

@Maheshpal Singh Rathore
@mpsrathore2020

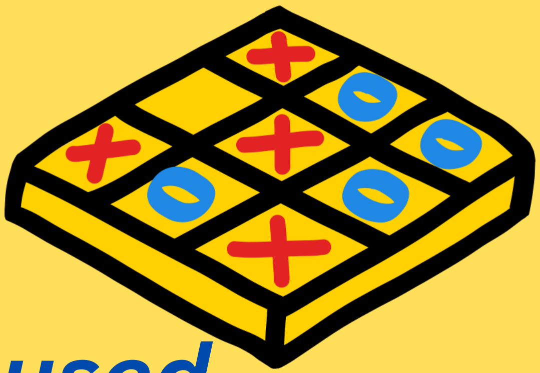
SWIPE >>>

There are three different Asymptotic notations-

- ***Big O Notation***
- ***Omega Notation***
- ***Theta Notation***

SWIPE >>>

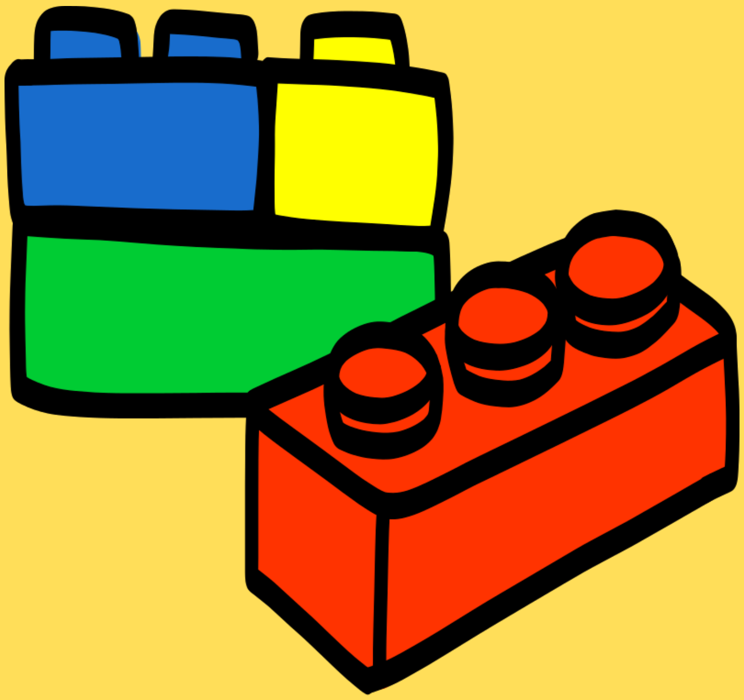
Big O notation (O) :



- *This is the most commonly used asymptotic notation, and describes the upper bound or worst-case scenario for the algorithm's time complexity.*

- *It expresses the growth rate of the algorithm's time complexity in terms of a function, and ignores constant factors and lower-order terms.*

- *For example, if an algorithm's time complexity is O(n^2), it means that its time complexity grows no faster than n^2, which is a quadratic function.*



Omega notation (Ω) :

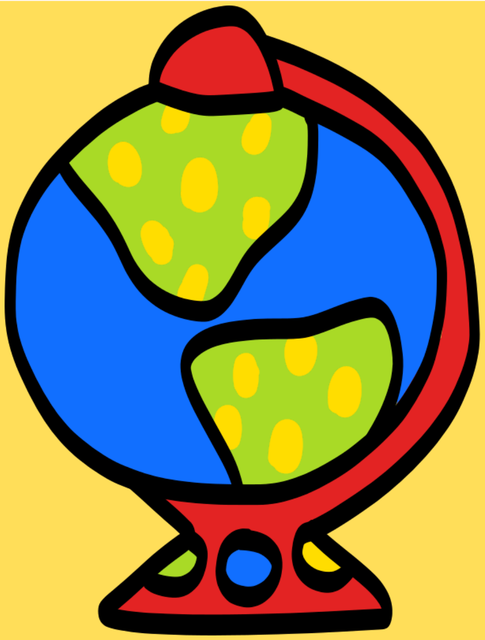
- *This describes the lower bound or best-case scenario for the algorithm's time complexity.*

- *It expresses the lower bound of the growth rate of the algorithm's time complexity in terms of a function.*

@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

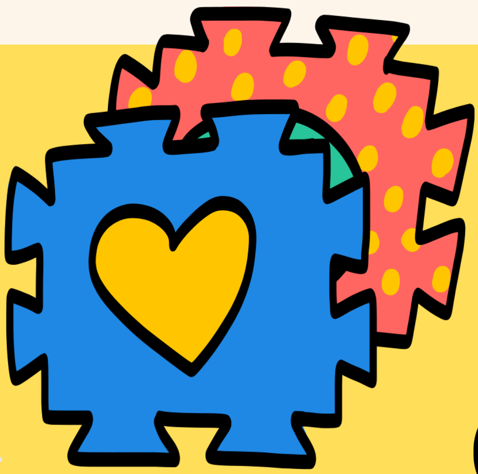
- ***For example, if an algorithm's time complexity is Ω(n^2), it means that its time complexity grows at least as fast as n^2.***



Theta notation (Θ):

- ***This describes the tight bound or average-case scenario for the algorithm's time complexity.***

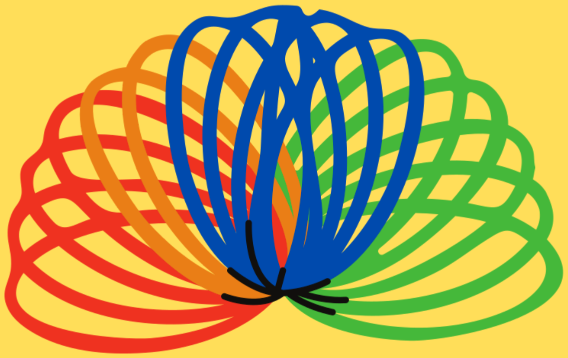
- ***It expresses both the upper and lower bounds of the growth rate of the algorithm's time complexity in terms of a function.***

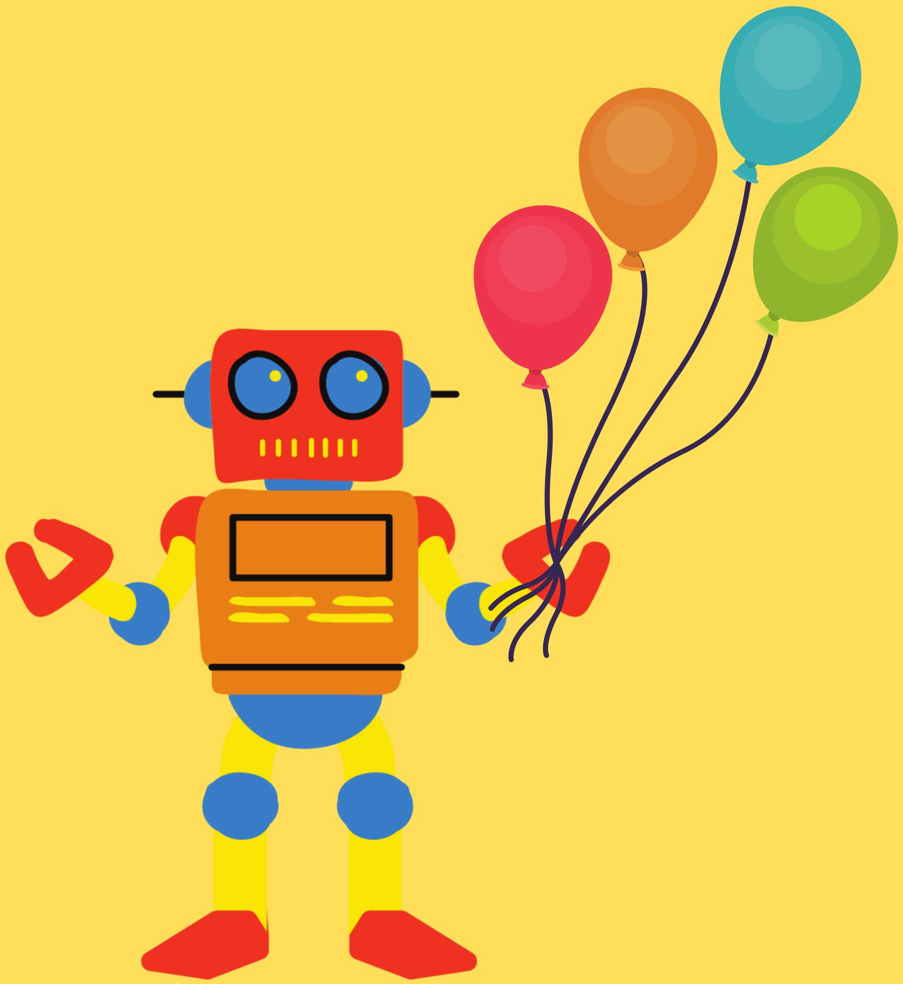


@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

- ***For example, if an algorithm's time complexity is $\Theta(n^2)$, it means that its time complexity grows exactly as fast as n^2 .***

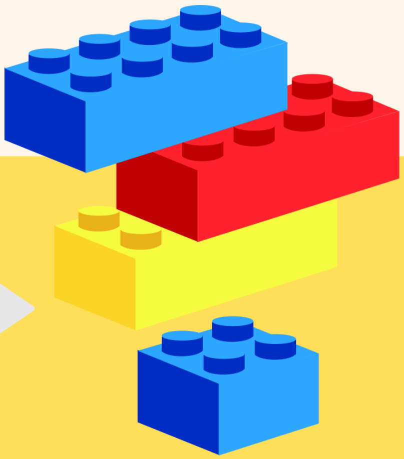




@Maheshpal Singh Rathore
@mpsrathore2020

- *These notations are useful for comparing the performance of different algorithms and determining which algorithm is best suited for a particular problem.*

- *They are also useful for predicting how the performance of an algorithm will change as the input size grows larger, which can help identify potential performance issues early in the development process.*



@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

Conclusion -

- ***Time complexity is important to consider because computers have limited processing power, and if an algorithm takes too long to run, it can cause problems like slow performance or unresponsive applications.***

- ***In general, it's best to design algorithms that are as efficient as possible, while still accomplishing the desired task.***

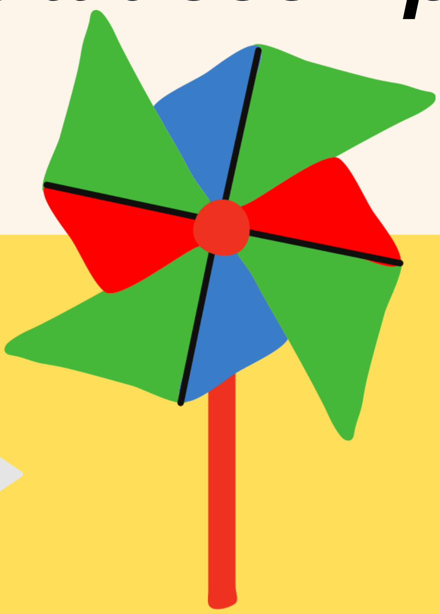
@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

Conclusion -

- *Space complexity is important to consider because computers have limited memory, and if an algorithm requires too much memory to run, it can cause problems like crashes or slow performance.*

- *In general, it's best to design algorithms that use as little memory as possible, while still accomplishing the desired task.*



@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

Do they depend on system ?

- *The time and space complexity of an algorithm do not depend on the system where the algorithm is executed.*
- *Instead, they depend only on the algorithm itself and the size of the input data.*

@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

- ***For example, consider the bubble sort algorithm .***

- ***The time complexity of this algorithm is $O(n^2)$, which means that the number of steps required to sort the data increases quadratically as the size of the data increases.***

- ***This relationship between the number of steps and the input size is independent of the system where the algorithm is executed.***

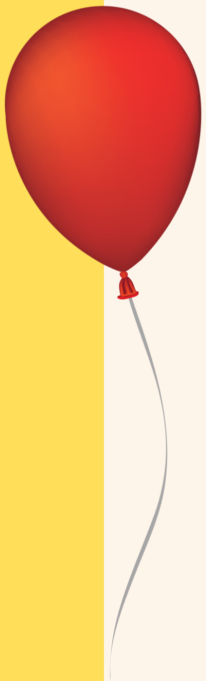
@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

- ***Similarly, the space complexity of an algorithm describes how much memory is required to store the input data and any additional memory required by the algorithm itself, and is also independent of the system where the algorithm is executed.***

@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>



- ***However, the actual runtime of an algorithm can be affected by the system where it is executed, due to factors such as the processing power of the CPU, the amount of available memory, and the speed of the input/output devices.***

- ***Nonetheless, the time and space complexity of the algorithm itself remain constant, regardless of the system where it is executed.***

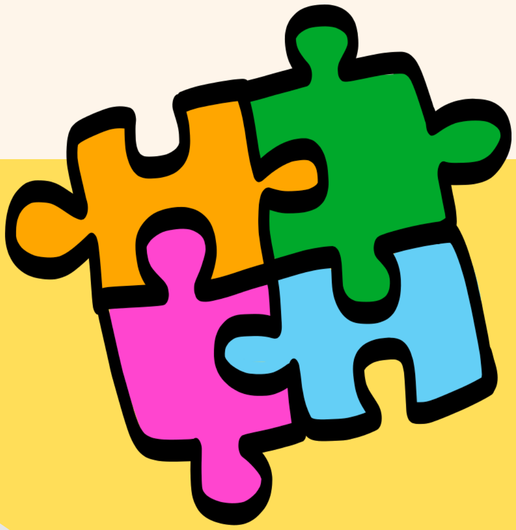
@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

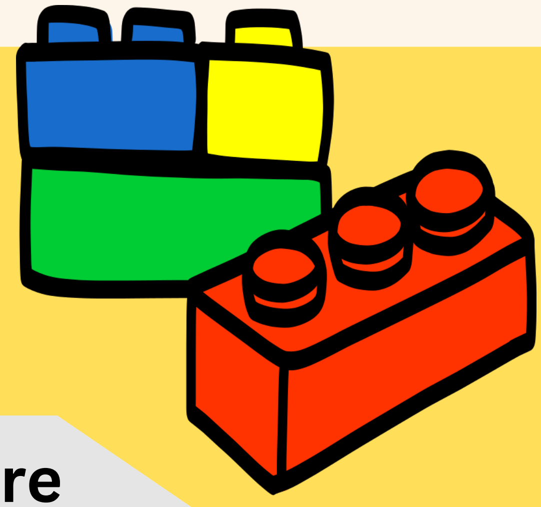
SUMMARY :

- *Time complexity describes how the runtime of an algorithm grows as the input size increases .*

- *Space complexity describes how the memory requirements of an algorithm grow as the input size increases.*



- ***Real-world examples of these concepts can be found in any situation where you need to analyze the performance of an algorithm, such as optimizing database queries, improving website load times, or designing efficient data structures for a large-scale software project.***



@Maheshpal Singh Rathore
@mpsrathore2020

SWIPE >>>

DID YOU FIND THIS POST HELPFUL ?

Follow us on

@Maheshpal Singh Rathore

@mpsrathore2020

@Kiran Kanwar Rathore

@kiranrathore123

Like , Save, and Share with your friends !!!

Credit - Internet