Welcome back to the

Java Course

Module 2

Computational Complexity

Computational complexity concerns how much time and how much memory a computer takes to solve a problem.

In other worlds, the efficentness of an algorithm.

Computational Complexity

It is essential in computer science because it helps us analyze and compare algorithms, predict their behavior on large inputs, and make informed decisions about which algorithm to use in practical applications.

Computational Complexity

Let’s suppose we want to match couples of cards.

Computational Complexity

They are hidden and we can reveal one at a time.

Computational Complexity

What is the faster way to find the twin cards?

Computational Complexity

We know that an algorithm is a sequence of steps aimed to get a result or to solve a problem.

But we can reach the same result through infinite algorithms.

Computational Complexity

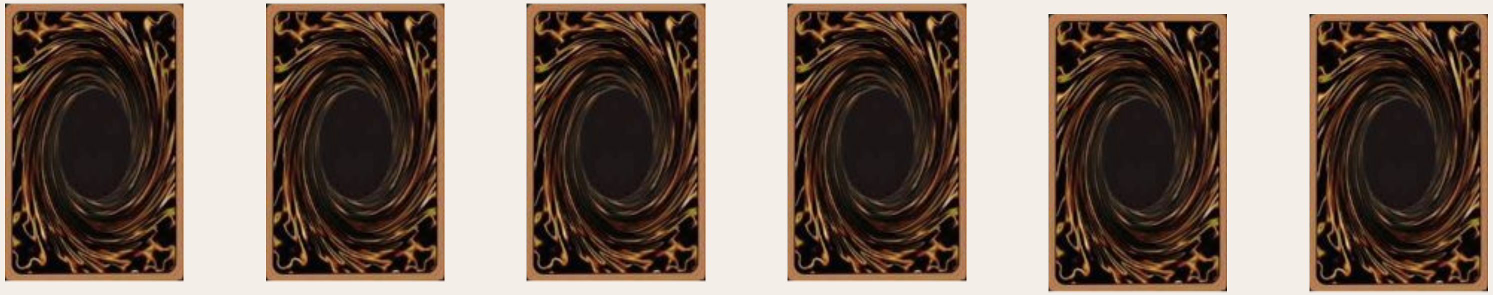
The computational complexity is the relationship between the time and the resources needed.

Often a linear relationship is the best method.

Computational Complexity

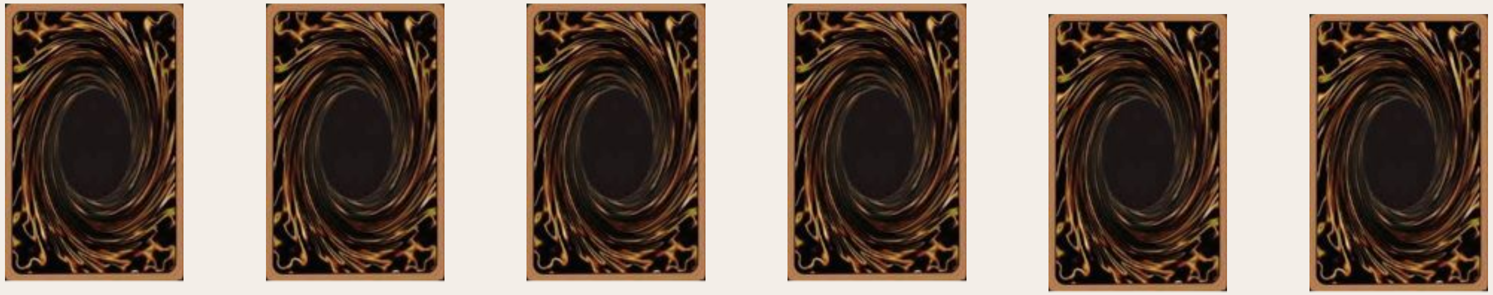
In the previous example, what is the maximum number of cards we have to stock in memory before we find a match?

How much time do we need?



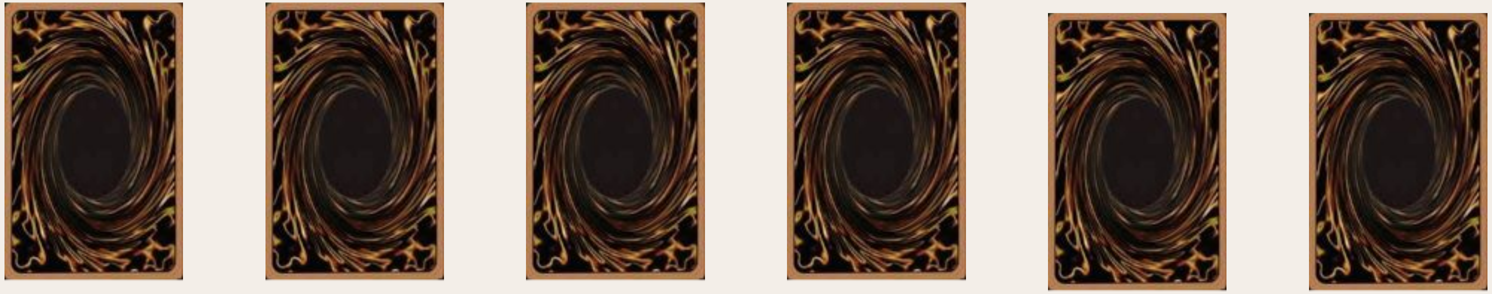
Computational Complexity

If we have 12 cards, we can examine all the cards and store their values in 12 variables. Then, to find a match, we might randomly select two variables for comparison. Probably this solution has the maximum cost of resources and time.



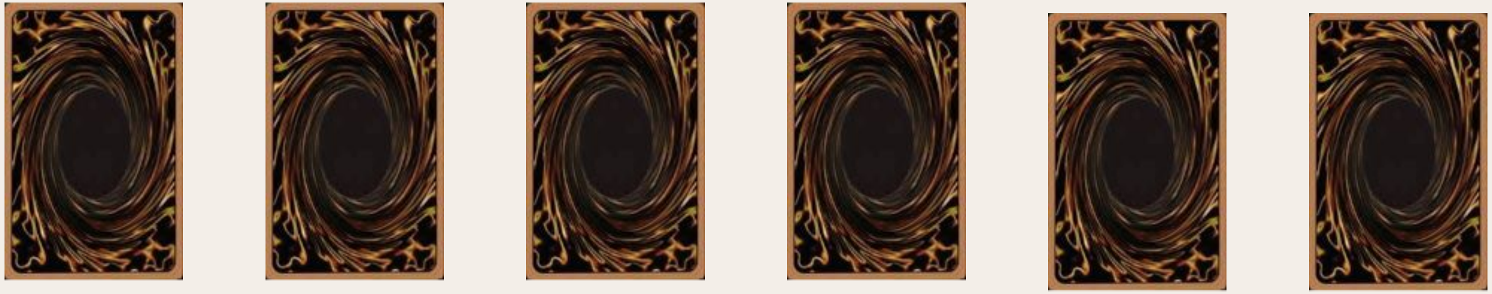
Computational Complexity

Even in the worst case, 6 variables are enough. We can check the match each time we examine a new card. In the worst case, after the sixth card we'll always find a match. This is definitely a method implemented better than the previous one.



Computational Complexity

We can also store only the value of the first card and examine the others looking for the match with the first. Then, after the match is found, we repeat the process. This solution needs time, but is good if we don’t know the number of cards.



Practice!

Practice

Now you can use functions and data structures to simplify your code.

Let's use them in real examples!

Are they easier if you use functions?

Practice

You've recently acquired a small shop and need to develop a program to assist the cashier in performing calculations. The cashier can input product prices and specify the percentage of any potential discounts, but you'll handle the actual calculations!

Practice

You are the proud owner of an all-you-can-eat sushi restaurant! Now you need to manage customer orders. Display the menu to customers, and they will enter the dish code. However, please note that they cannot enter more than 10 dishes in a single order. They can, however, place unlimited orders!

Practice

You are the owner of a second-hand bookshop that buys and resells used books. Customers can sell you as many books as they want, and the book’s value is evaluated based on the number of pages, publication date and condition. Create a program that calculates the final amount to offer for the purchase. Then, print on the screen the customer's name and the titles of the books he has sold.