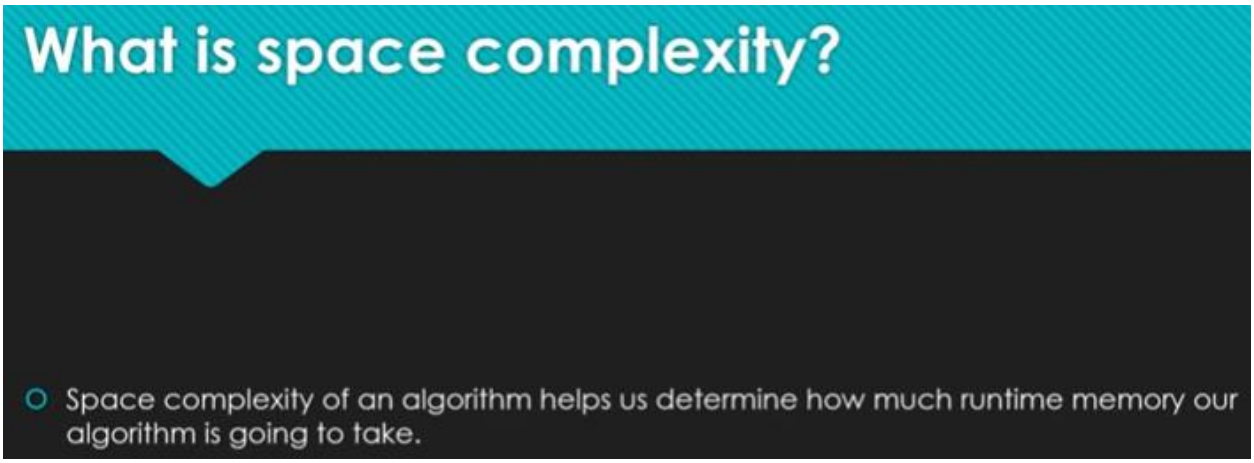

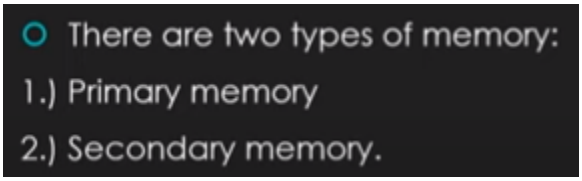

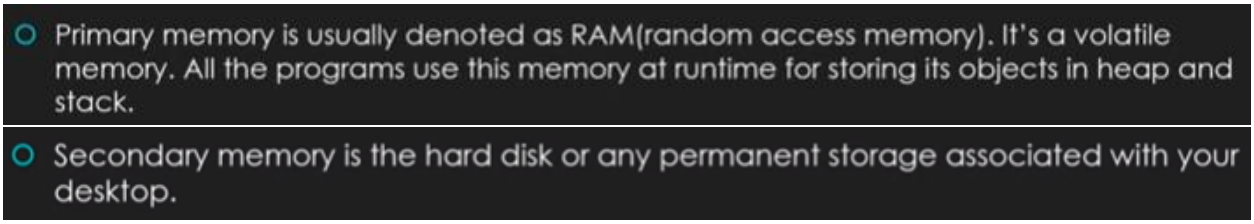

RAM is more volatile.. If we are working on a word document, we need to periodically save it... otherwise, all our work could be lost. Once we save our work, it will be stored in the hard disk.

Space Complexity versus Time Complexity

Space Complexity: How much more **memory use (RAM)** do we need as the inputs provided to the Code gets larger?

Time Complexity: How much more **runtime** do we need as the inputs provided to the Code gets larger?

Both of the above still use Big O Notation

Space Complexity Rules of Thumb

Storing values in **variables** always takes up memory.

Most Primitives (Booleans and Numbers) take up O(1) / Constant Space.

var x = 100 & var x = 200 take up **same** amount of memory

Strings, Arrays, and Objects take up O(n) / Linear Space

An Array w/ 4 elements takes up **twice** the memory of Array w/ 2 elements.

Why is Time Complexity Prioritized Over Space Complexity?

Costs to **produce** and **run** processors are much **higher** compared to RAM.

--When a strenuous program is running, the fans are running to cool the processors, not the RAM.

--The processors are doing the strenuous work, and they use way more electricity than RAM.

Consumers / Users in general care more about speed than RAM usage.

-- Google Chrome takes up more RAM, but it's faster.

-- Facebook and instagram don't use much RAM, but if their posts and comments are slow to load, this will lead to poor user experience and they will lose users.

What is Space Complexity?

Space complexity is the **total amount of memory space used by an algorithm/program including the space of input values for execution**. So to find space complexity, it is enough to calculate the space occupied by the variables used in an algorithm/program.

But often, people confuse Space complexity with Auxiliary space. Auxiliary space is just a temporary or extra space and it is not the same as space complexity. In simpler terms,

> **Space Complexity = Auxiliary space + Space use by input values**

Example #1

```
#include<stdio.h>
int main()
{
  int a = 5, b = 5, c;
  c = a + b;
  printf("%d", c);
}
```

Output:

Explanation: Do not misunderstand space complexity to be 1364 Kilobytes as shown in the output image. The method to calculate the actual space complexity is shown below.

In the above program, 3 integer variables are used. The size of the integer data type is 2 or 4 bytes which depends on the compiler. Now, lets assume the size as 4 bytes. So, the total space occupied by the above-given program is 4 * 3 = 12 bytes. Since no additional variables are used, no extra space is required.

Hence, **space complexity for the above-given program is O(1), or constant.**

Summary

Big O Notation	Space Complexity details
O(1)	Constant Space Complexity occurs when the program doesn't contain any loops, recursive functions or call to any other functions.
O(n)	Linear space complexity occurs when the program contains any loops.