DATA STRUCTURES & ALGORITHMS

Algorithm's Complexity

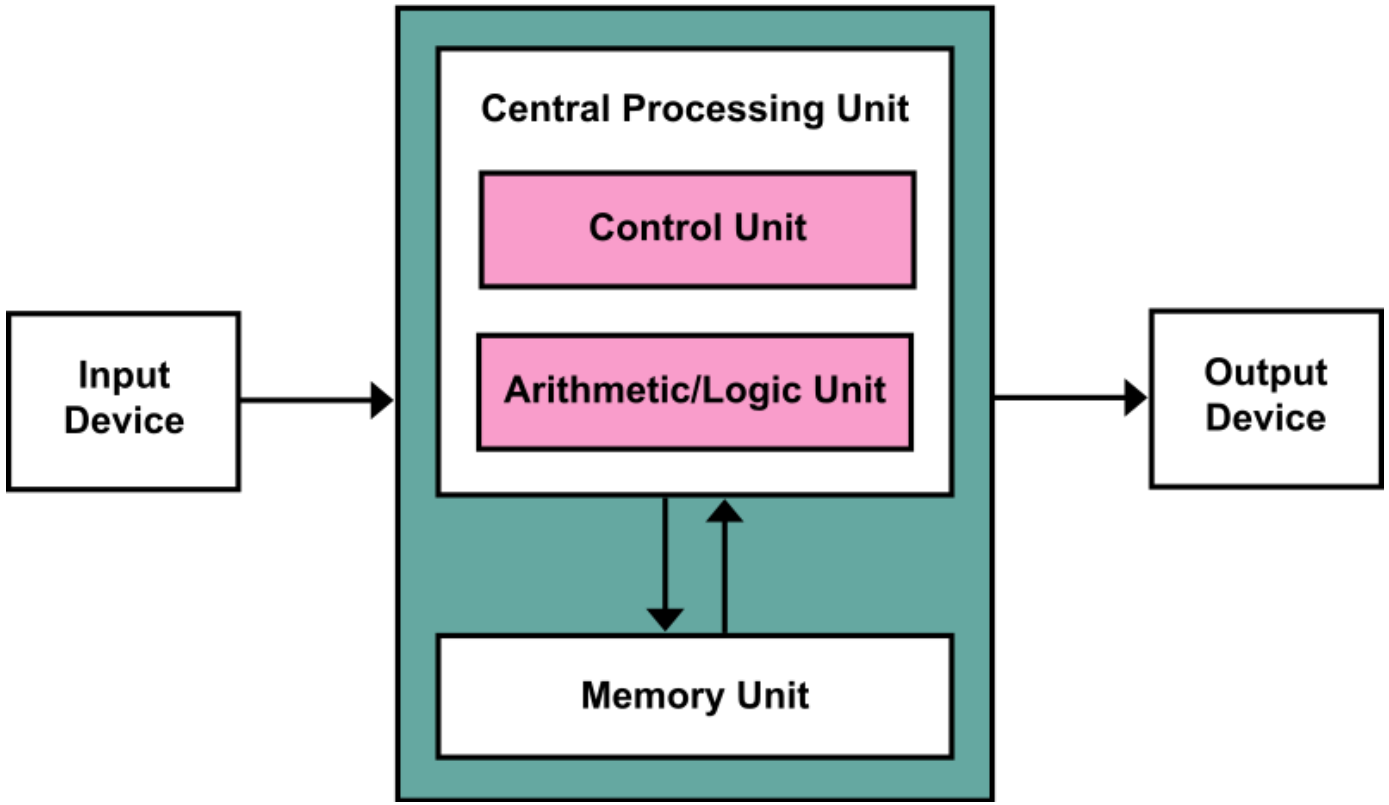
Instructor: Engr. Laraib Siddiqui

Computing Runtime

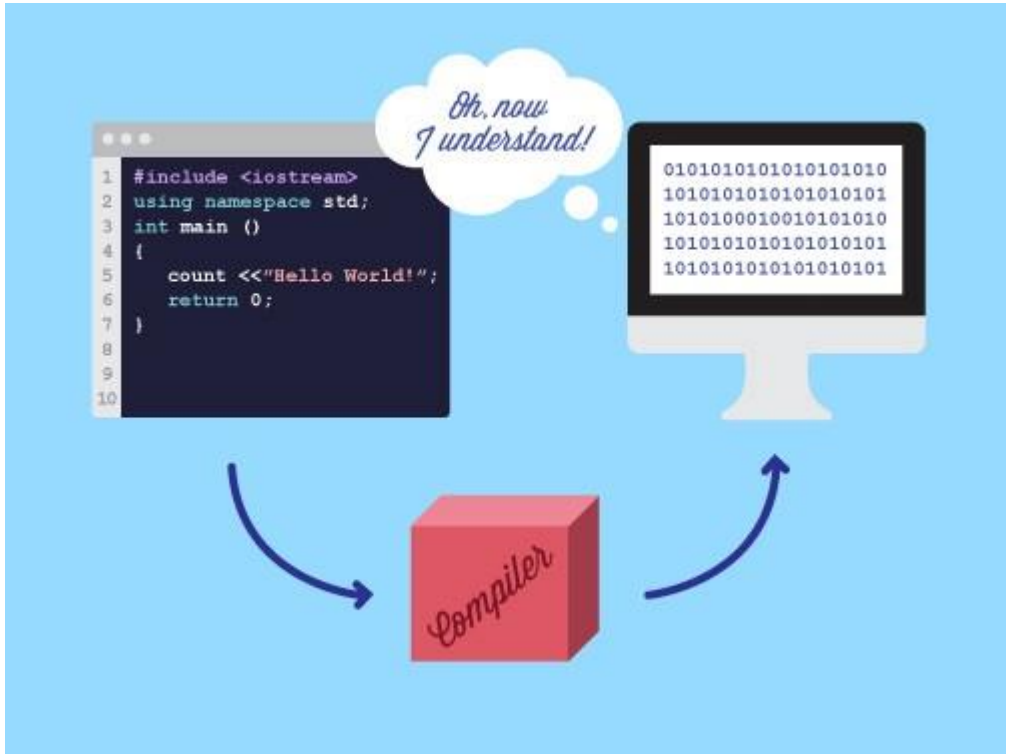
To figure out how long this simple program would actually take to run on a real computer, we would also need to know things like:

Computer Speed

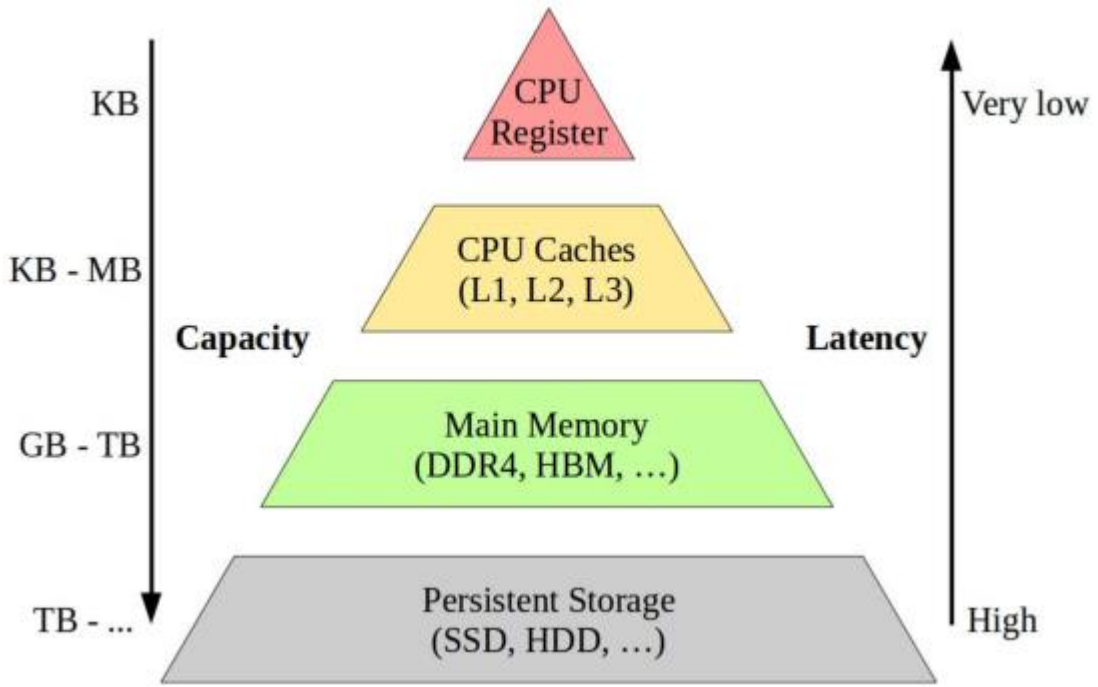
Computer Architecture



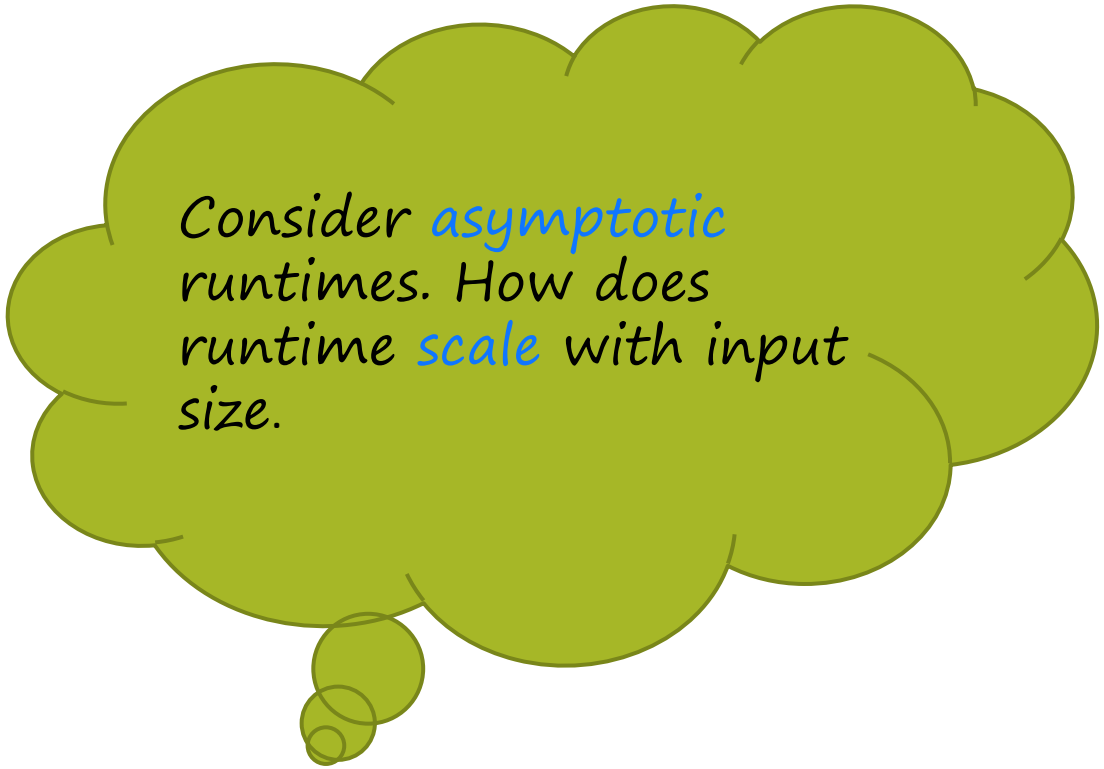
Compiler


Oh, now I understand!
#include <iostream>
using namespace std;
int main ()
{
count <<"Hello World!";
return 0;
}
0101010101010101010
1010101010101010101
1010100010010101010
1010101010101010101
1010101010101010101
Compiler

Memory Hierarchy



Issue



In practice, you might not even know some of these details.

Asymptotic Notation

Asymptotic Notation is used to describe the running time of an algorithm - how much time an algorithm takes with a given input, n.

Algorithm's Complexity

- Function *f(n)* which measures the time and space used by an algorithm in terms on input size *n*.
- Way to classify how efficient an algorithm is compared to alternative ones.
- Computational complexity and efficient implementation depends on suitable data structures.

Space Complexity

- Amount of memory consumed by the algorithm (apart from input and output, if required by specification) until it completes its execution.
- The space occupied by the program is generally by the following:
  - ✓ A fixed amount of memory occupied by the space for the program code and space occupied by the variables used in the program.
  - ✓ A variable amount of memory occupied by the component variable whose size is dependent on the problem being solved. This space increases or decreases depending upon whether the program uses iterative or recursive procedures.

Space Complexity

- Memory taken by the instructions is not in the control of the programmer as its totally dependent upon the compiler to assign this memory.
- Memory space taken by the variables is in the control of a programmer. More the number of variables used, more will be the space taken by them in the memory.
- Three different spaces considered for determining the amount of memory used by the algorithm.
- Instruction space
  - ✓ memory occupied by compiled version of program.
- Data space
  - ✓ hold variables, data structures, allocated memory and other data elements.
- Environmental space
  - ✓ memory uses on the run time stack for each function call.

Time Complexity of an Algorithm

Amount taken by an algorithm to run.

- Compilation time
  - ✓Time taken to compile (checking syntax & semantic errors, linking to libraries) an algorithm.
- Run Time:
  - ✓Time to execute the compiled program.
  - ✓Depend upon the number of instructions present in the algorithm.

NOTE: Run time is calculated only for executable statements and not for declaration statements.

Time Complexity of an Algorithm

Time complexity of an algorithm is generally classified as three types.

1. Worst case
   - ✓ Longest time that an algorithm will use to produce desired result.
   - ✓ Maximum value of f(n) for any possible input.
2. Average Case
   - ✓ Average time (average space) that an algorithm will use to produce desired result
   - ✓ Expected value of f(n)
3. Best Case
   - ✓ Lowest time that an algorithm will use to produce desired result