Class 01: Basic Introduction

There two types of data types: Primitive, Derived, User defined
https://ide.geeksforgeeks.org/NS1pF5Bib6
Primitive: int, double, float, char,bool
Address of a variable: printf("%p",(void*)&n); [Here n is a variable]
Bool: 1byte
C99 has bool. Library function is <stdbool.h>. Standard states that bool values behave as integral types, yet it doesn't specify their concrete representation in memory.
No format specifier for C but cin works fine for C++.
Char: 1byte

    Input: %c Output: %c

Integer:

short int => 2bytes
Input: %hi Output: %d/%hi
(unsigned)short int => 2bytes
Input: %hu Output: %d/%hu

```
short int a; unsigned short int b;
scanf("%hi %hu",&a,&b);
printf("%hi %hu\n",a,b);
printf("%d %d\n",a,b);
```

===

int => 4bytes
Input: %d Output: %d
unsigned int => 4bytes
Input: %u Output: %u

```
int a; unsigned int b;
scanf("%d %u",&a,&b);
printf("%d %u\n",a,b);
```

===

long long int=>8bytes
Input: %lld Output: %lld
unsigned long long int=>8bytes
Input: %llu Output: %llu

```
long long int a; unsigned long long int b;
scanf("%lld %llu",&a,&b);
```

```
            printf("%lld %llu\n",a,b);
```

```
#include <limits.h>
#include <float.h>
```

Float: 4byte (6decimal places) 1.2E-38 to 3.4E+38
Input: %lf Output: %f
Double: 8byte (15decimal places) 2.3E-308 to 1.7E+308
Input: %lf Output: %f/%lf(version wise might not work)
Long Double: 16byte (19decimal places) 3.4E-4932 to 1.1E+4932
Input: %lf Output: %Lf

```
    float f; double d; long double ld;
    scanf("%f %lf %Lf",&f,&d,&ld);
    printf("%f %f %Lf",f,d,ld);
```

%*c,%E/%e(scientific notation for float r double),%X/%x(Hexadecimal),%o(octal)

Input/ Output Format:

File input/output:
Easiest=>

```
FILE * freopen ( const char * filename, const char * mode, FILE * stream );
 freopen("input.txt", "r", stdin);
 freopen("output.txt", "w", stdout);
```

https://www.guru99.com/c-file-input-output.html
https://codeforces.com/problemset/problem/234/B

OJ Verdicts:
https://icpc.kattis.com/help/judgements#:~:text=Output%20Limit%20Exceeded%20means%20that,it%20flooded%20our%20hard%20drive
Pending/in queue/processing: this is actually not a verdict, it's shown when the code is submitted and yet not completed judging.
Accepted: When everything is going right
Presentation Error: Presentation error means: The data in your output is correct, but it is not formatted in the proper way. Check the problem statement. (Missing/excessive blank lines and unnecessary spaces are likely to have caused this message.)
Wrong Answer: Data they want doesn't match yours
Compilation Error: Compile Error means that we failed to compile your source code. In order to help you debug the error. Syntax error.
Runtime Error:

1. Math Error
2. Non-existent memory
3. Segmentation Error

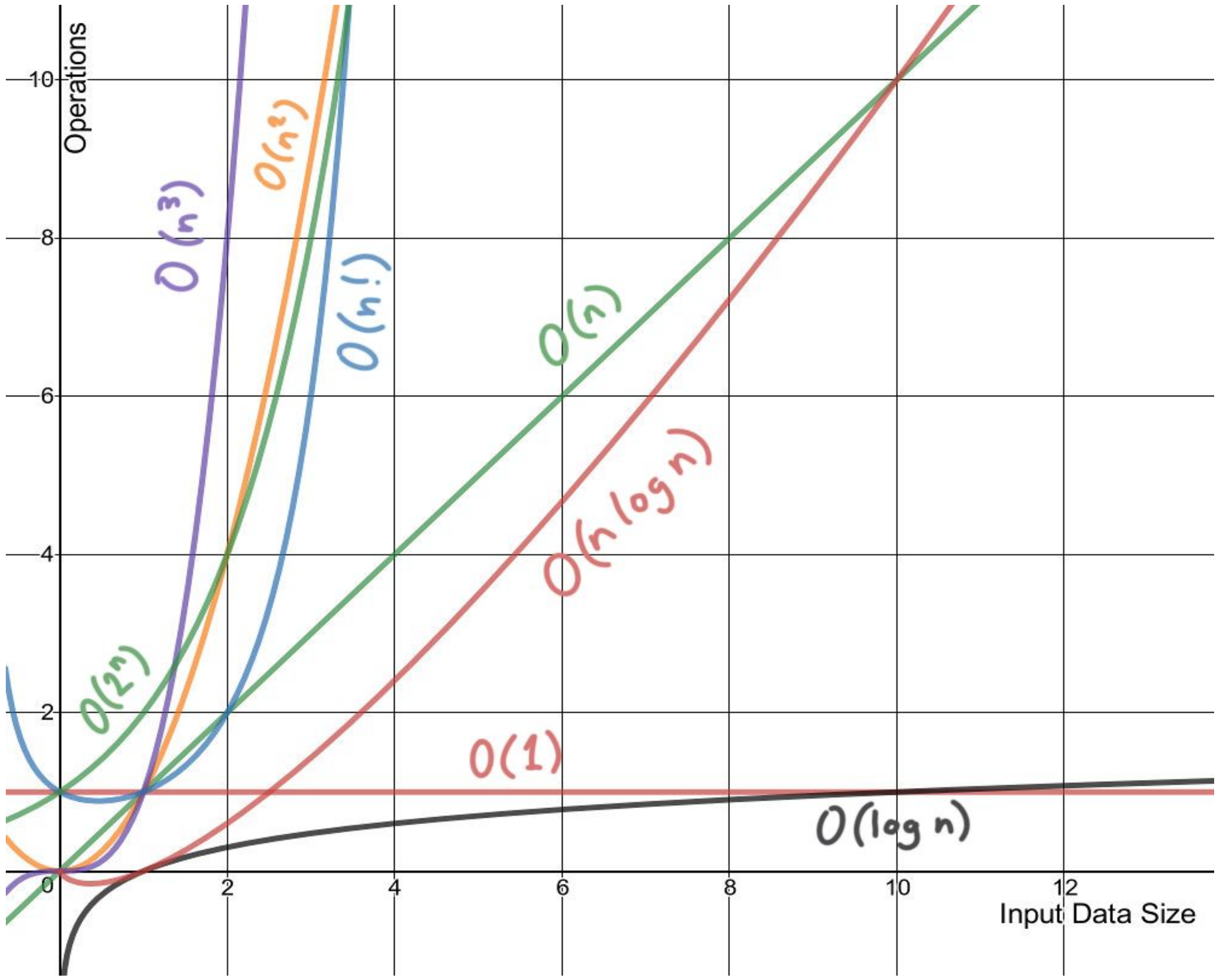
4. & percent cause try to access the memory that is not permitted to.This means that your program has tried to access memory which it was not allowed to access, either because the memory was not mapped by the process or due to permission errors.

Memory Limit Exceeded: You used more memory than you are given.
Output Limit Exceeded: Output Limit Exceeded means that your program has produced too much output and we decided to shoot it down before it flooded our hard drive.
Time Limit Exceeded: To be continued…

[Bug Finding and Tracing:]

Use all possible cases
Use random cases
Step by step going
Bangla debugging.

[Time Complexity & Optimization:]



Best Case: O(1) bogus
Worst Case:
Average Case:
https://www.geeksforgeeks.org/analysis-of-algorithms-set-2-asymptotic-analysis/

Blog: http://www.shafaetsplanet.com/?p=1313
https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/?fbclid=IwAR1e34B_YPbQOPI_TSvhxzOiCx6N69rtVRZnLWKprigyM7pD8B4a3zN7ERg#O-2-n-Exponential-time
Videos: https://www.youtube.com/watch?v=9TlHvipP5yA
https://www.youtube.com/watch?v=9SgLBjXqwd4
https://www.youtube.com/watch?v=p1EnSvS3urU
For optimization:
1st think of a bruteforce solution then try to notice which part is impossible to optimize then try to calculate the time complexity of every subtask then check if it is possible to optimize the complexity of the subtask
Problem Solving Strategy, Guideline and CP:
=> Taking notes on the key points of the problem
=> Step by step flowchart
=> Observation & analytical growth
=> Monthly Practice target

N number er list
K ekta number
K er anagram gula amr happy number
Definition anagram
Q ta question
Prottek question e range dibe which is x y
Prottek question e x theke y porjonto joto gula number ache tar moddhe happy number koyta seta ber korbo…

0 1 0 1 1
=> 10^5*120=1.2*10^7+10^5
ps[1]=0
ps[2]=ps[1]+1=1 =(a to b)
ps[3]=ps[2]+0=1=a+b+c
ps[4]=ps[3]+1=2
ps[5]=ps[4]+1=3

=>Ekta question 10^5=((10^5)/10^8)=0.
=>for(x to y) 1 to n
=>Current k er anagram func(curr,k)
=>If current number ta happy=>
=>Output
{
Curr k,
}
5 arr[5];

A b c d e
1->a ar[0]
2->a+b arr[0]+arr[1]
3->a+b+c
4->a+b+c+d
5->a+b+c+d+e
3 5
(1 to y) - (1 to x-1)= (x to y)
ps[5]-ps[2]=2

File I/O:
https://vjudge.net/problem/Gym-100283G

freopen(“jenga.in”,”r”,stdin);
freopen(“output.txt”,”w”,stdout);
A_1=sin(1)
A_2=sin(1-sin(2))
A_3=sin(1-sin(2+sin(3)))
A_4=sin(1-sin(2+sin(3-sin(4))))

$S_n = (...(A_1+n)A_2+n-1)A_3+...+2)A_n+1$
S_1 = sin(1)+1
S_2=(sin(1)+2)sin(1-sin(2))+1
$S_3=((A_1+3)A_2+2)A_3+1$
$S_4=(((A_1+4)A_2+3)A_3+2)A_4+1$

Class 02: Function and Recursion

1) Time Complexity
2) Function Overview
3) Parameter and Return type with example
4) Loop vs recursion
5) Recursion exercises

Class 03: Data structures and Introduction to C++

1) STL
2) Containers
3) Data structures

Class 04: Number Theory Part - 1

1) Base Conversion
   a) Binary to Decimal (Vice versa)
   b) Hexadecimal To Binary (Vice Versa)
2) GCD & LCM (Log_2N)
3) Modular arithmetic
4) Exponent And Logarithms
5) Factors and Divisor
6) Bigmod

Class 05: Number Theory Part - 2

1) Formula Progression
2) Primality check (optimized)
3) Factorization (optimized)
4) Sieve

Class 06: Searching and Sorting

1) Linear Search
2) Binary Search
3) $O(N^2)$ Sorting
4) $O(Nlog_2N)$ Sorting (Theory)

Class 07: Basic Graph + String

1) String Matching $O(N^2)$ & Palindrome
2) Graph representation

3) Traversal (BFS/DFS)