Competitive Programming

Introduction:

- for including all packages like “stdio.h”, “math.h” and others in one include you just need to use :

```
#include <bits/stdc++.h>
```

- In compile time you can use following flags:
  - std=c++11 for forcing g++ to compile with this version
  - O2 will optimize your code
  - Wall will show all warnings (sometimes it is helpful for example when you have used some variable in wrong places.)
  - Wfatal-errors will help you to just show only one error in compile time. (Sometimes too much errors can be annoying

```
g++ -std=c++11 -O2 -Wall -Wfatal-errors
```

- Don’t Remember to use long long in cases you have a variable for summation of int variables. overflow is a common mistake.

- Look at this framework. Build a framework for coding to be as fast as possible in coding. It’s just my framework! as a sample :

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef vector<int> vi;
typedef pair<int , int> pi;
typedef tree<int,null_type,less<int>,rb_tree_tag,tree_order_statistics_node_update> in-
dexed_set;
template <typename T> using index_set =  tree<T, null_type, less<T>, rb_tree_tag, tree_or-
der_statistics_node_update>;

#define N_INFLL (numeric_limits<ll>::max() + 1)
#define P_INFLL (numeric_limits<ll>::max())
#define N_INFI (numeric_limits<int>::max() + 1)
#define P_INFI (numeric_limits<int>::max())
#define F first
#define S second
#define PB push_back
#define MP make_pair
#define LB lower_bound
#define UB upper_bound
#define REP(i,a,b) for (int i = a; i <= b; i++)
#define Loop(i,n) for (int i = 0; i < n; i++)
#define SQ(a) (a)*(a)
#define InputI(a) int a; cin >> a;
#define InputLL(a) long long a; cin >> a;
#define InputS(a) string a; cin >> a;
#define InputAI(a , n) int a[n]; for (int i = 0; i < n; ++i) cin >> a[i];
#define InputALL(a , n) long long a[n]; for (int i = 0; i < n; ++i) cin >> a[i];
#define InputS(a) string a; cin >> a;

int main(int argc, char const *argv[])
{
        ios::sync_with_stdio(0);
        cin.tie(0);

        return 0;
}
```

Binary Search:

- look at these two implementation of binary search and think how they work:

```
int a = 0;
int b = n-1;
while (a<=b){
        int k = (a+b)/2;
        if (array[k]==x)
                return k;
        else if (array[k] < x)
                a = k+1;
        else
                b = k-1;
}
```

```
int k = 0;
for (int b = n/2; b>=1 ; b/=2)
        while(k+b < n && array[k+b] <= x)
                k+=b;
if (array[k]==x)
        return k;
```

- c++ libraries also have functions for binary search. You can search a little about them to understand how they work. they are lower_bound , upper_bound , equal_range:
- look at these codes and tell what they do :
  - ```
    auto k = lower_bound(array,array+n,x);
    if (k < n && array[k] == x)
            return k;
    ```
  - ```
    auto a = lower_bound(array, array+n , x) - array;
    auto b = upper_bound(array, array+n , x) - array;
    return (b-a);
    ```
  - ```
    auto r = equal_range(array, array+n , x);
    return (r.second - r.first);
    ```

you can ask any of them in class if you didn't understand what they do.
    ```

- two famous questions can be answered with binary search:
  - smallest solution : Suppose that we wish to find the smallest value k that is a valid solution for a problem.

```
int x = -1;
for (int b = z; b >= 1; b /= 2)
        while (!ok(x+b))
                x += b;
int k = x+1;
```

  - maximum value : find the maximum value for a function that is first increasing and then decreasing.

```
int x = -1;
for (int b = z; b >= 1; b /= 2)
        while (f(x+b) < f(x+b+1))
                x += b;
int k = x+1;
```

- Now Answer these questions for training:
  You don’t have to submit code. Just don’t be disappointed if you can’t solve. Think of algorithm and idea and write anything you found about questions on a paper. We will discuss on questions if we had time in class:

  https://cses.fi/problemset/task/1142/
  https://cses.fi/problemset/task/1085/
  https://cses.fi/problemset/task/1086/
  https://cses.fi/problemset/task/1091/