{ } Unionville HS

⚠️REMINDER: ⚠️
PAY YOUR SAC FEE

Computer Club 2024-25

Competitive Programming

Lesson #3 (Time/Space Complexity)!

Google Classroom Code: slwprdp

Time Complexity

- Way of measuring the speed of a program.

Space Complexity

- Way of measuring the memory of a program.

- Useful for checking if an idea will pass given constraints before implementation.
- Analyzes WORST CASE SCENARIO
- Generally constants are ignored.

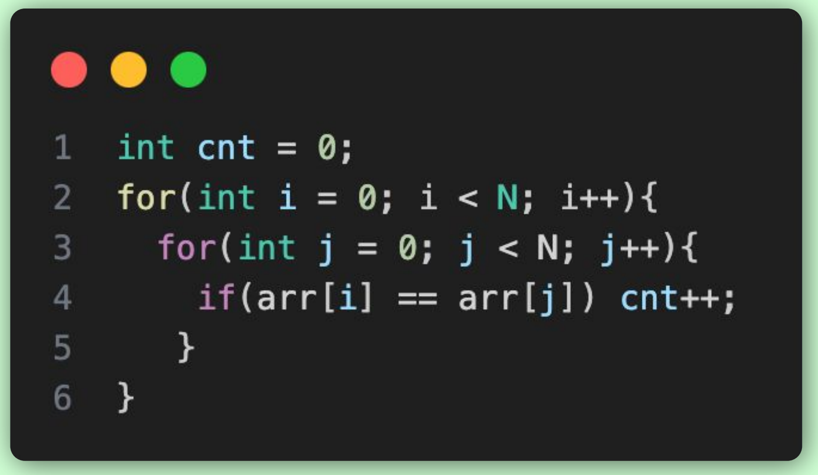
Big O Notation

- Way of representing time & space complexity.
- We ignore constants

Examples:

- O(N) -> N operations = O(N + 1)
- O(N^2) -> N^2 operations = O($2N^2$)
- O(logn) -> log (base 2) n operations

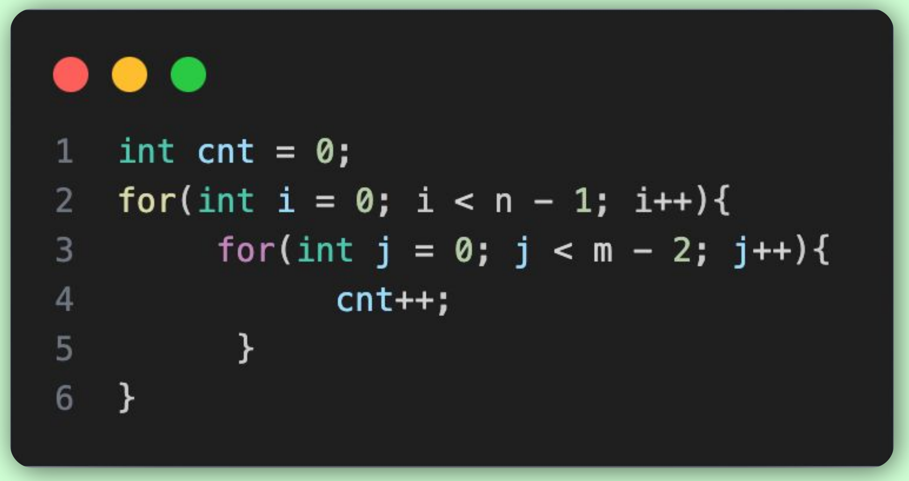
Try It Yourself!



```
int cnt = 0;
for(int i = 0; i < N; i++){
  for(int j = 0; j < N; j++){
    if(arr[i] == arr[j]) cnt++;
  }
}
```

Answer: $O(N^2)$

Try It Yourself!



```
int cnt = 0;
for(int i = 0; i < n - 1; i++){
     for(int j = 0; j < m - 2; j++){
           cnt++;
      }
}
```

Answer: O(NM)

Useful Time Complexities

$O(N!)$: $N <= 10$

$O(2^N)$: $N <= 20$

$O(N^5)$: $N <= 50$

$O(N^4)$: $N <= 100$

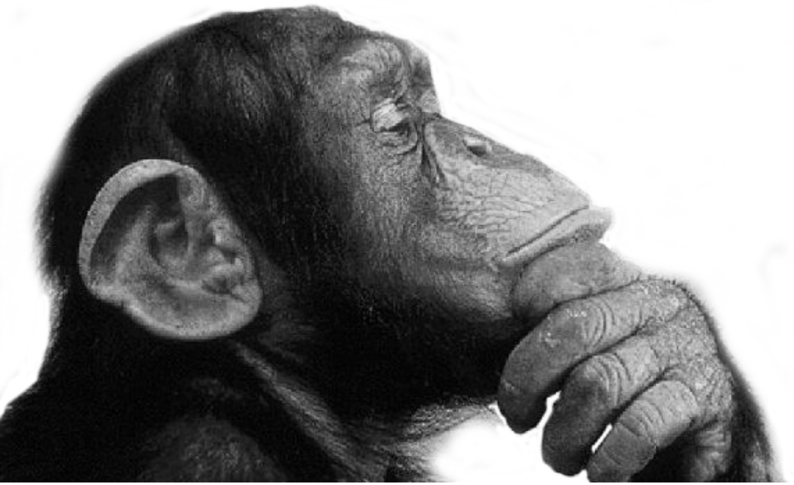
$O(N^3)$: $N <= 500$

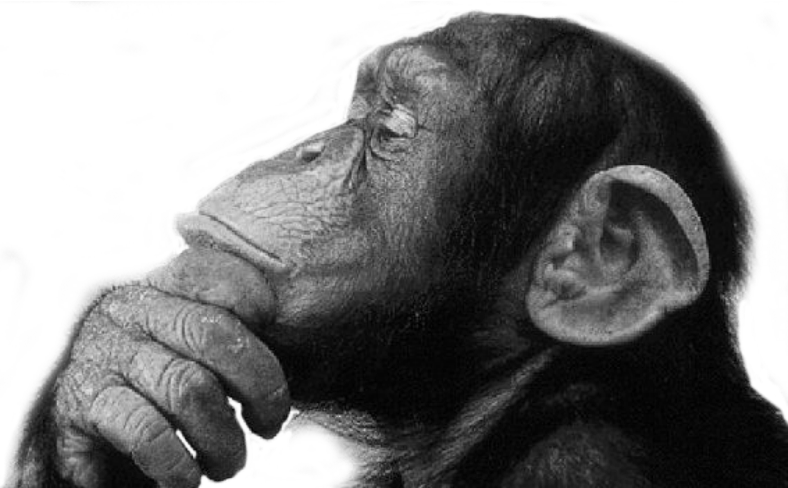
$O(N^2 \log N)$: $N <= 3000$

$O(N\log N)/O(N\sqrt{N})$: $N <= 10^5$

$O(N)/O(N\sqrt{N})$: $N <= 10^6$

CHALLENGE PROBLEM





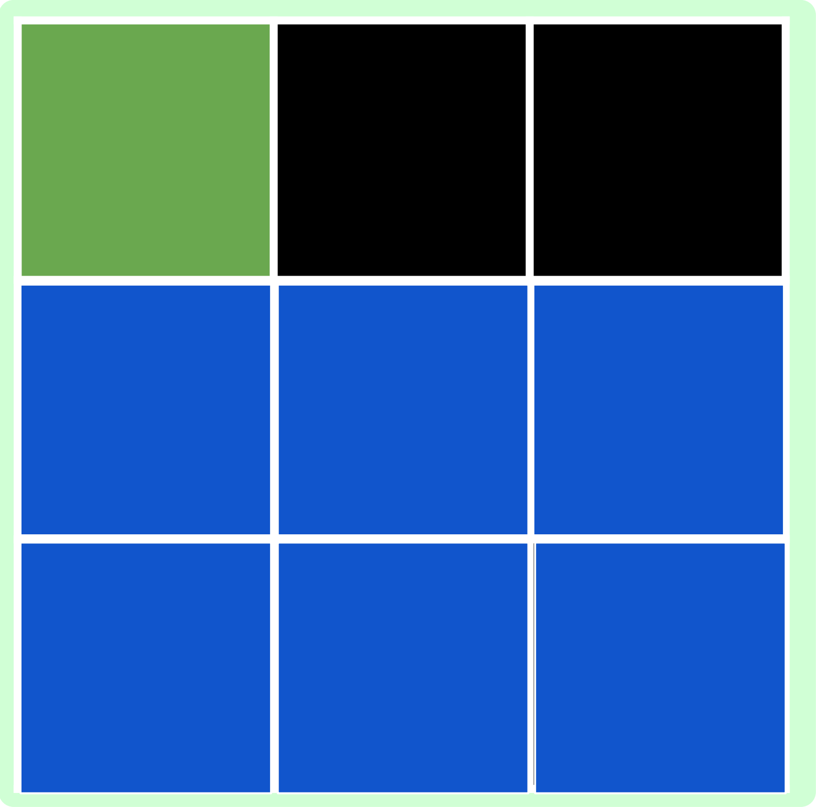
N is very large we cannot have time or space complexity of $O(N^2)$!

What do we do?

Use T!

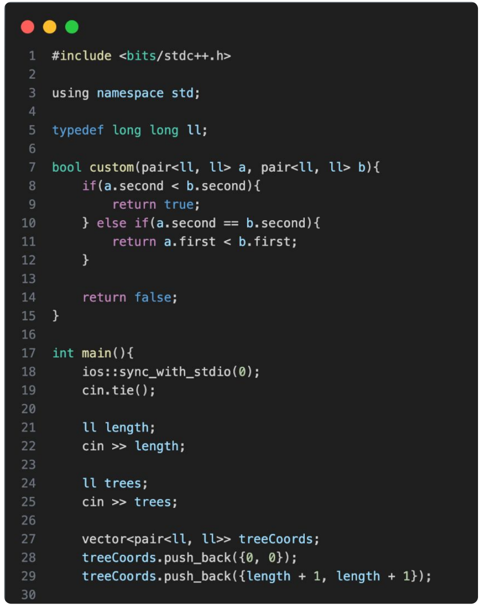
How can we solve the problem using T?

Observation: Every pool is limited by adjacent borders or trees.

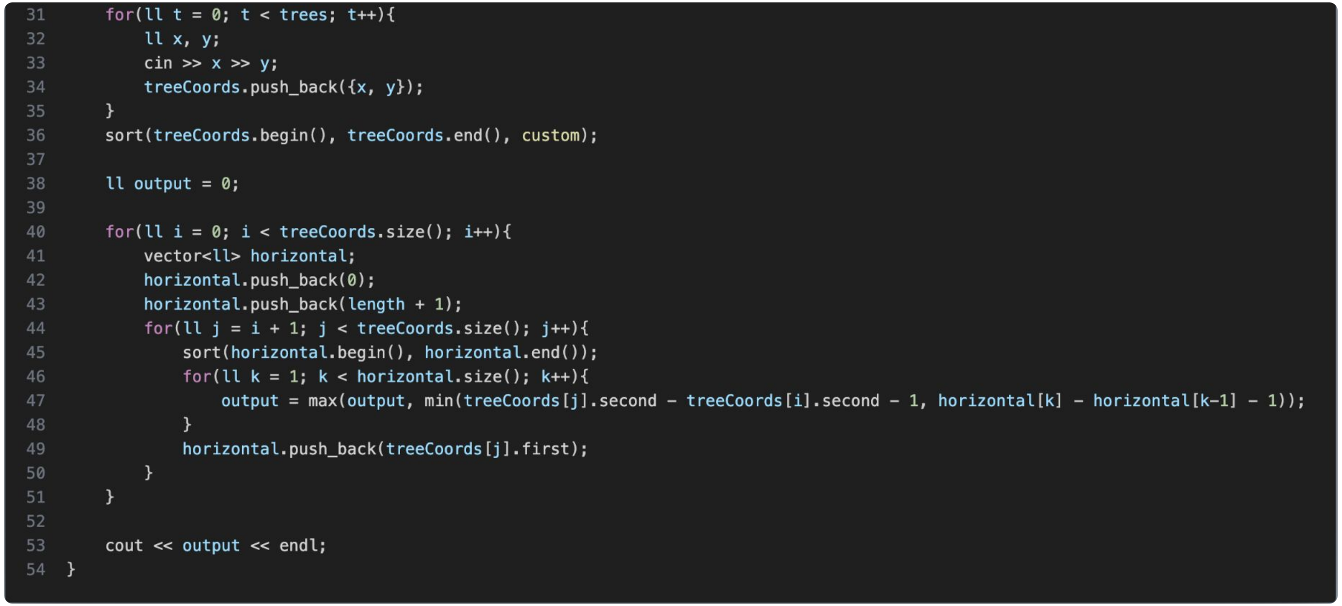


The Main Idea

Since each pool is limited by a tree/border on the top, bottom, left, and right. We can loop through all possibilities since the number of trees is quite small. We can ‘split’ the entire grid into subrectangles based on the number of trees. The answer is then the maximum square in each of these subrectangles.

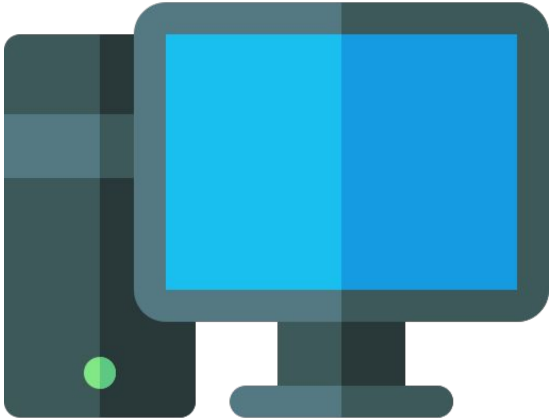


```cpp
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

bool custom(pair<ll, ll> a, pair<ll, ll> b){
    if(a.second < b.second){
        return true;
    } else if(a.second == b.second){
        return a.first < b.first;
    }

    return false;
}

int main(){
    ios::sync_with_stdio(0);
    cin.tie();

    ll length;
    cin >> length;

    ll trees;
    cin >> trees;

    vector<pair<ll, ll>> treeCoords;
    treeCoords.push_back({0, 0});
    treeCoords.push_back({length + 1, length + 1});

```



```cpp
    for(ll t = 0; t < trees; t++){
        ll x, y;
        cin >> x >> y;
        treeCoords.push_back({x, y});
    }
    sort(treeCoords.begin(), treeCoords.end(), custom);

    ll output = 0;

    for(ll i = 0; i < treeCoords.size(); i++){
        vector<ll> horizontal;
        horizontal.push_back(0);
        horizontal.push_back(length + 1);
        for(ll j = i + 1; j < treeCoords.size(); j++){
            sort(horizontal.begin(), horizontal.end());
            for(ll k = 1; k < horizontal.size(); k++){
                output = max(output, min(treeCoords[j].second - treeCoords[i].second - 1, horizontal[k] - horizontal[k-1] - 1));
            }
            horizontal.push_back(treeCoords[j].first);
        }
    }

    cout << output << endl;
}
```

Homework Practice



Medium

Medium

Medium

Hard

Bye 👋

Please Leave…

Instagram: **unionvillecs2024**

Discord: **QWBUScbMzz**

Classroom code: **slwprdp**