

Algorithm And Problem Solving Lab 15B17CI471

Project Report

Title : BATTLE STRATEGY

Submitted By :

Ashish Chauhan	9919103062
Aniket Saraswat	9919103090
Ritik Arora	9919103087
Pankaj Kumar	9919103077
Himanshu Sharma	9919103066

Submitted To :

Dr. Shilpa Budhkar
Dr. Akanksha Mehndiratta



Department of CSE/IT
Jaypee Institute of Information Technology, Noida
May 2021

Statement :

A program for the design of the army's arrangement, according to the position and arrangement of the opposition army in the battlefield (which we know with the help of Satellite images). From, arrangement of the army, we meant the arrangement of troops with different abilities within the army.

Motivation behind the topic :

Motivation behind choosing this project is very simple, the group members never ending love towards the army. Most of us have actually tried for the NDA Exam, and none of us could get through. So we feel like contributing to the army in any way possible, and we feel this could be one opportunity for the same.

Objective :

The objective of this project is to help the army with a simple technological method to make an instant pre-battle strategy which includes the positioning of troops according to their abilities, for example if by the satellite images we know that the troops with heavy armours are in the frontline of opposition army, so it will be foolish if we don't arrange our army accordingly, if we do it all manually, then it won't be this smooth, and might require a lot of time.

Methodology :

Let's look at a picture representation for a better view,

Opponent's Army Arrangement

A	A	A	A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D	D	D	D
D	D	D	D	D	D	D	D	D	D	D	D	D	D

A,B,C,D = Opponent's Troop, with their attacking powers given to us

Our Army unarranged

W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z
W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z
W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z
W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z
W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z
W	W	X	X	X	Y	Y	Y	Y	Y	Z	Z	Z	Z

W,X,Y,Z = Army's troop, with their attacking power given to us

In above example we can conclude that opponent's army on paper looks like, or the figures we can make out by looking at the satellite pictures are as follow :

Troop's Type	Power of artillery	No. of troops	Total power
A	100	14	1400
B	250	28	7000
C	50	14	700
D	10	28	280

And the resources we have available with us are :

Troop's Type	Power of artillery	No. of troops	Total power
W	110	12	1320
X	20	18	360
Y	350	30	10500
Z	100	24	2400

The above given are the figures which the army strategist will give as an input to the system, and the information about the arrangement will be returned back as outputs.

Program :

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

class Troops{
public:
    int troopsnum;
    int artillery;
    int power;
};

int Lower_bound(vector<Troops> &arr, int N, int X)
{
    int mid;
    int low = 0;
    int high = N;

    while (low < high) {
        mid = low + (high - low) / 2;
        if (X <= arr[mid].power) {
            high = mid;
        }
        else {
            low = mid + 1;
        }
    }
    if(low < N && arr[low].power < X) {
        low++;
    }
    return low;
}

bool compare(Troops a, Troops b){
    return a.power < b.power;
}

int main(){
    int n,n2;
    vector<Troops> arr,arr2,result;
    cout<<"Enter the number of Troops of army A: \n";
    cin>>n;
    cout<<"\nEnter the Troops number and Artillery of army A: \n";
    for(int i = 0; i<n; i++){
        Troops T;
```

```

    cin>>T.troopsnum>>T.artillery;
    int p = (T.troopsnum)*(T.artillery);
    T.power = p;
    arr.push_back(T);
}
sort(arr.begin(), arr.end(), compare);
cout<<"\nEnter the number of Troops of army B: \n";
cin>>n2;
cout<<"\nEnter the Troops number and Artillery of army B: \n";
for(int i = 0; i<n2; i++){
    Troops T;
    cin>>T.troopsnum>>T.artillery;
    int p = (T.troopsnum)*(T.artillery);
    T.power = p;
    arr2.push_back(T);
}
for(int i = 0; i<arr2.size(); i++){
    if(arr.size() == 0)
        break;
    Troops T = arr2[i];
    int index = Lower_bound(arr,arr.size(),T.power);
    if(index >= arr.size()){
        cout<<"Army A cannot win this battle in any formation.";
        return 0;
    }
    result.push_back(T);
    int leftpower = arr[index].power - arr2[i].power;
    if(leftpower > 0 && arr.size() > 1){
        Troops T2 = arr[arr.size() - 1];
        T2.power += leftpower;
        arr.pop_back();
        arr.push_back(T2);
        arr.erase(arr.begin() + index);
    }
    else if(arr.size() == 1 && arr2[i].power <= arr[0].power){
        arr[0].power = arr[0].power - arr2[i].power;
    }
}
cout<<"\n";
cout<<"The formation of Army A for winning against Army B should be :";

for(int i = 0; i<result.size(); i++){
    cout<<result[i].power<<" ";
}

```

```
return 0;  
}
```

```
Enter the number of Troops of army A:  
5  
  
Enter the Troops number and Artillery of army A:  
2 5  
6 5  
2 3  
4 5  
6 8  
  
Enter the number of Troops of army B:  
8  
  
Enter the Troops number and Artillery of army B:  
4 5  
6 2  
2 3  
1 2  
7 4  
3 1  
1 6  
4 2  
  
The formation of Army A for winning against Army B should be  
:20 12 6 2 28 3 6 8
```

```
Enter the number of Troops of army A:  
5  
  
Enter the Troops number and Artillery of army A:  
2 5  
6 5  
2 3  
4 5  
6 8  
  
Enter the number of Troops of army B:  
8  
  
Enter the Troops number and Artillery of army B:  
4 5  
6 2  
2 3  
1 2  
7 4  
3 1  
1 6  
4 2  
Army A cannot win this battle in any formation.
```

Contribution :

The biggest contribution we can say here is, that it could save the lives of army personnel and moreover it will save a lot of time in pre-battle preparations. Also it will help in avoiding the minute human errors, which are generally most expected when there is a huge battle pressure on the head of the strategist.