

## Lab1 Solutions

---

### Min Max

ประกาศตัวแปรสองตัว ตัวแปรแรกเพื่อเก็บค่าต่ำสุด(ค่าเริ่มต้นควรมากที่สุด) และตัวแปรที่สองเพื่อเก็บค่าสูงสุด(ค่าเริ่มต้นควรน้อยที่สุด) จากนั้นวนลูปทั้งอาร์เรย์เพื่อเปรียบเลขตัวนั้น ๆ กับค่าต่ำสุดและค่าสูงสุด ถ้าต่ำกว่าค่าต่ำสุดปัจจุบันให้เปลี่ยนค่าต่ำสุดเป็นเลขนั้น ถ้าสูงกว่าค่าสูงสุดปัจจุบันให้เปลี่ยนค่าสูงสุดเป็นเลขนั้น

### Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin >> n;

    int arr[n];
    for(int i=0;i<n;i++) cin >> arr[i];

    int min=10001,max=-10001;
    for(int i=0;i<n;i++)
    {
        if(arr[i] < min) min = arr[i];
        if(arr[i] > max) max = arr[i];
    }

    cout << min << endl << max;
}
```

*time complexity :  $O(n)$*

## Wiring

สามารถติดตั้งสายไฟระหว่างเสาไฟฟ้าต้นที่  $i$  และต้นที่  $j$  ได้ก็ต่อเมื่อระหว่างเสาไฟฟ้าทั้งสองต้นนี้ไม่มีเสาไฟฟ้าต้นใดเลยที่สูงมากกว่าหรือเท่ากับเสาไฟฟ้าต้นที่  $i$  หรือต้นที่  $j$  ดังนั้นควรวนลูปเช็คกว่าเสาต้นที่  $i + 1$  ถึง  $j - 1$  สูงน้อยกว่าทั้งเสาต้นที่  $i$  และ  $j$

### Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n,ans=0;
    cin >> n;

    int arr[n];
    for(int i=0;i<n;i++) cin >> arr[i];

    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            bool check = true;
            for(int k=i+1;k<j;k++)
                if(arr[k] >= min(arr[i],arr[j])) check = false;
            if(check == true) ans++;
        }
    }
    cout << ans;
}
```

*time complexity :  $O(n^3)$*

## Ann Orchard

คำนวณหาทุกพื้นที่สี่เหลี่ยมที่เป็นไปได้ แล้วหาผลบวกเลขทั้งหมดภายในสี่เหลี่ยมแต่ละแบบ คำตอบคือผลบวกที่มากที่สุด (คำตอบจะไม่น้อยกว่า 0) ในโค้ดตัวอย่างจะวนลูป  $i_1, j_1, i_2, j_2$  แล้วคำนวณผลบวกเลขทุกจำนวนจากตำแหน่ง  $(i_1, j_1)$  ถึง  $(i_2, j_2)$  ผ่านลูป  $x$  และ  $y$

### Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n,ans=0;
    cin >> n;

    int arr[n][n];
    for(int i=0;i<n;i++) for(int j=0;j<n;j++) cin >> arr[i][j];

    for(int i1=0;i1<n;i1++)
    {
        for(int j1=0;j1<n;j1++)
        {
            for(int i2=i1;i2<n;i2++)
            {
                for(int j2=j1;j2<n;j2++)
                {
                    int sum = 0;
                    for(int x=i1;x<=i2;x++)
                    {
                        for(int y=j1;y<=j2;y++) sum+=arr[x][y];
                    }
                    ans = max(ans,sum);
                }
            }
        }
    }
    cout << ans;
}
```

time complexity :  $O(n^6)$