Nearest Smaller to Right

Problem Statement:

Given an array of integers, find the closest (not Considering distance, but value) Smaller on right of every element. If an element has no Smaller on the right side print -1.

arr: | 4 | 5 | 2 | 10 | 8 |

i i i i i

i 2 -1 8 -1

- check for Smaller on right to i if Smiler then take that else take -1.

Brute force:

```
for (int i=0; i<n-1; i++)
    for ( j = i+1 ; j<n ; j++)
```

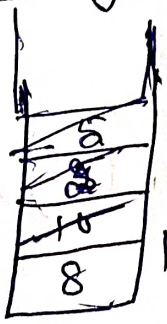
j is related to i

→ If you are traversing through loop in forward direction (L-R) then it will be more Time Consuming else Go through (R-L) to reduce time

Complexity:

arr: 4 5 2 10 8

4 → checks in right



Stack pop

| -1 | 8 | -1 | 2 | 2 |

reverse ⇒ reverse the vector

change :- ① Right to left traversal.
② Smaller
③ vector reverse.

```
vector<int> v;
stack<int> s;
for (int i=size-1; i>=0; i--)
{
    if(s.size()==0)
    {
        v.push_back(-1);
    }
    else if( s.size()>0 && s.top()<arr[i])
    {
        v.push_back(s.top())
    }
    else if(s.size()>0 && s.top>=arr[i])
    {
        while(s.size>0 && s.top()>=arr[i])
        {
            s.pop();
        }
        if(s.size()==0)
            v.push_back(-1)
        else
            v.push_back(s.top())
    }
    s.push_back(arr[i])
    reverse(s.begin, s.end());
}
```