```
/*

                    Name: Mehul Chaturvedi
                    IIT-Guwahati

*/

/*
                    PROBLEM STATEMENT
There is a company named James Peterson & Co. The company has
employees.
The employees have skills from 0 to n-1. All the employees have
distinct skills.
The manager of James Peterson & Co. wants to sort the employees
on the basis of their
skills in ascending order. He is only allowed to swap two
employees which are adjacent to each
other. He is given the skills of employees in an array of size .
He can swap the skills as
long as the absolute difference between their skills is 1. You
need to help the manager out and
tell whether it is possible to sort the skills of employees or
not.
Input Format:
First Line will have an integer  denoting the no. of test cases.
First line of each test case contains an integer  denoting the
no. of employees in the company.
Second line of each test case contains  distinct integers in the
range [0, n-1].
Output Format:
For each test case, print if it is possible to sort the skills
otherwise .
Constraints:
1 <= t <= 10
1 <= n <= 10^5
Sample Input:
2
4
1 0 3 2
3
2 1 0
Sample Output:
Yes
No
Explanation:
In first T.C., [1, 0, 3, 2] -> [0, 1, 3, 2] -> [0, 1, 2, 3]
In second T.C., [2, 1, 0] -> [1, 2, 0]  OR  [2, 1, 0] -> [2, 0,
1] So, it is impossible to sort.

*/


#include <bits/stdc++.h>

using namespace std;

bool skillSort(vector<int> &skills, int start, int end){
     // cout << start<<" " <<end<< '\n';
     if (start>= end)
     {
          return 1;
     }

     int n = skills.size();
     int mid = (start+end)/2;

     int left = skillSort(skills, start, mid);
     int right = skillSort(skills, mid+1, end);

     if (left!=1 || right != 1)
     {
          return 0;
     }

     if (skills[mid+1]>skills[mid])
     {
          return 1;
     }

     // cout << "mid "<<skills[mid] << '\n';
     // cout << "mid+1 "<<skills[mid+1] << '\n';

     if (skills[mid]-skills[mid+1]>1)
     {
          return 0;
     }

     if (skills[mid+1]<skills[mid])
     {
          swap(skills[mid], skills[mid+1]);
     }
     return 1;
```

```
}


int main( int argc , char ** argv )
{
     ios_base::sync_with_stdio(false) ;
     cin.tie(NULL) ;

     int t;
     cin>>t;

     while(t--){
          int n;
          cin>>n;
          //cout << "n "<<n << '\n';

          std::vector<int> skills;
          for (int i = 0; i < n; ++i)
          {
               int temp;
               cin>>temp;
               skills.push_back(temp);
          }

          if (skillSort(skills, 0, n-1)==1)
          {
               cout << "Yes" << '\n';
          }else{
               cout << "No" << '\n';
          }

     }



     return 0 ;



}
```