```cpp
/*
    Name: REPLES-X Codechef OCT Long 2020
    Copyright: https://www.codechef.com/OCT20A/problems/REPLESX/
    Author: Sounish Nath
    Date: 09-10-20 19:00
    Description: Simple but too tricy to find corner cases
*/

#include <bits/stdc++.h>

using namespace std ;
using arrays = vector<int>;

/*
4 9 7 0 8
0 4 7 8 9
x = 4, p = 3, k = 4
nums[k] = 8
nums[p] = 7
0 4 7 0 9
0 0 4 7 9
*/

void f(){
    int N, X, p, k ;
    cin >> N >> X >> p >> k;
    arrays nums(N);
    for(int i = 0; i < N; i++) {
        cin >> nums[i];
    }
    sort(nums.begin(), nums.end());
    // get the pivot elem index
    auto pivot = [&]() -> int {
        int inf = 2e9+18, index = -1;
        for(int i = 0; i < N; i++) {
        if(nums[i] == X){
            if(inf > abs(i - p)){
                inf = abs(i - p); index = i;
                }
            }
        }
        return index;
    };
    int index = pivot();
```

```
    int ans = 0 ;
    if(nums[index] != X) {
        nums[k-1] = X;
        sort(nums.begin(), nums.end()); ans++ ;
    }
    if(nums[p-1] == X) {
        cout << "minimum operations: " << ans << endl; return ;
    }else if((p < k && X > nums[p-1]) or (p > k && nums[p-1] > X)){
        cout << "minimum operations: " << -1 << endl; return ;
    }
    index = pivot() + 1;
    cout << "minimum operations: " << abs(p - index) + ans << endl; return ;
}

int main() {
    srand(time(NULL)) ;
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int T;
    cin >> T;
    while(T--){
        f();
    }
}

```