HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Applied Algorithm Lab

Max-distance Sub-sequence

ONE LOVE. ONE FUTURE.

Max-distance sub-sequence

- Given a sequence $a_1, \dots, a_N$
- Consider a subset of the sequence

    The distance of the subset is defined to be the minimum distance between two elements

- Find the subset of N given elements containing exactly C elements such that the distance is maximal.
- Example

stdin	stdout

1
5 3
1 2 8 4 9
	3

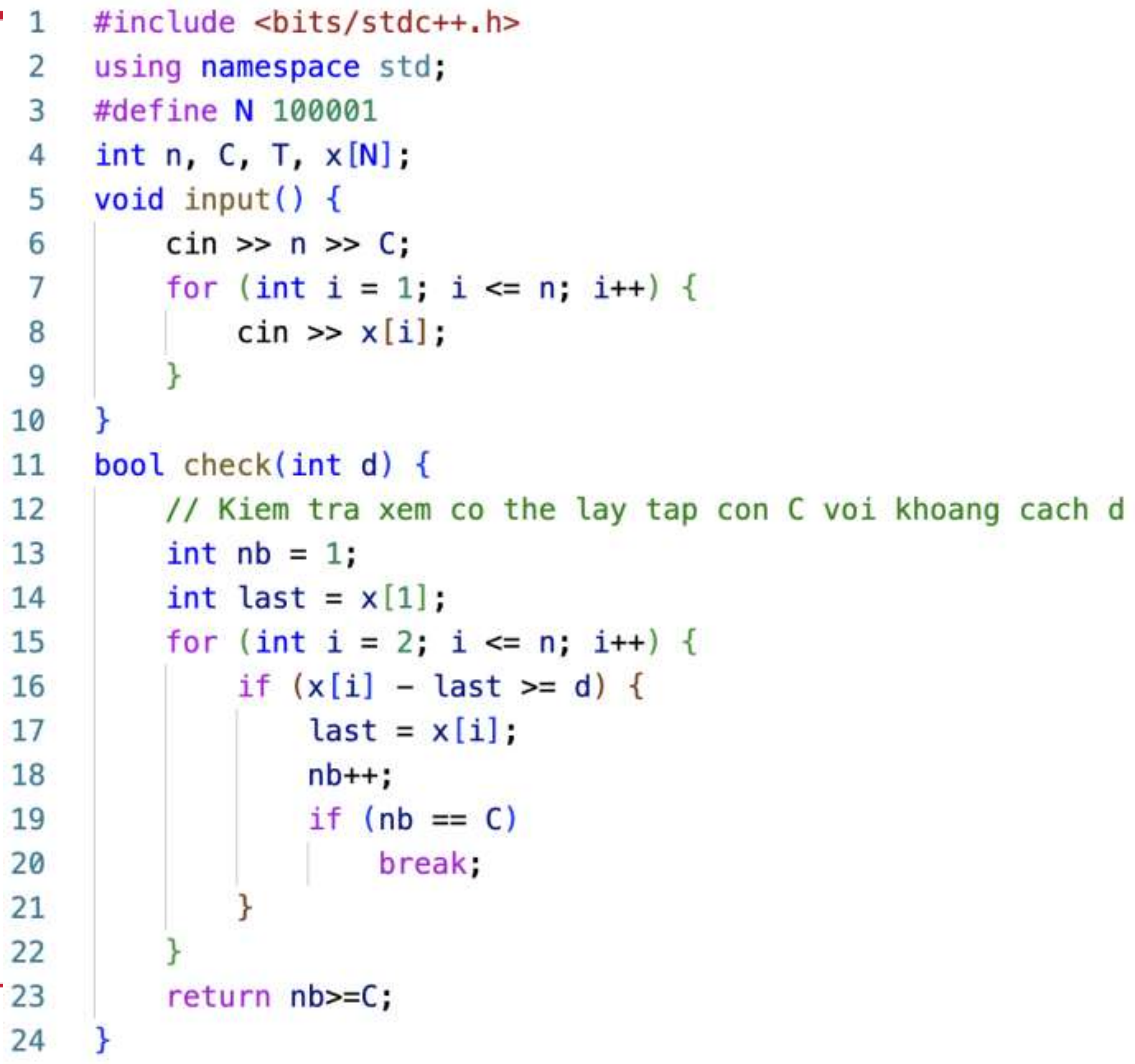
Max-distance sub-sequence

- Idea to solve: Sort $a_1, \ldots, a_N$ in increasing order
- The max-distance must be $\leq \frac{a_N - a_1}{c-1}$
- Find max-distance: traverse from $\mathrm{d} = \frac{a_N - a_1}{c-1}$ down to 1:
  - check(d): Check if we can find a sub-sequence with distance $\geq d$: Greedy
  - Add a1 into subsequence, use an auxiliary variable **last**
  - for i=2, i<=n; i++:
    - if a[i]-last ≥ d then we add a[i] to the subsequence and update **last**
    - otherwise continue
  - If the subsequence has C elements then d is max-distance. Print d
  - Complexity: O(n^2)

Max-distance sub-sequence

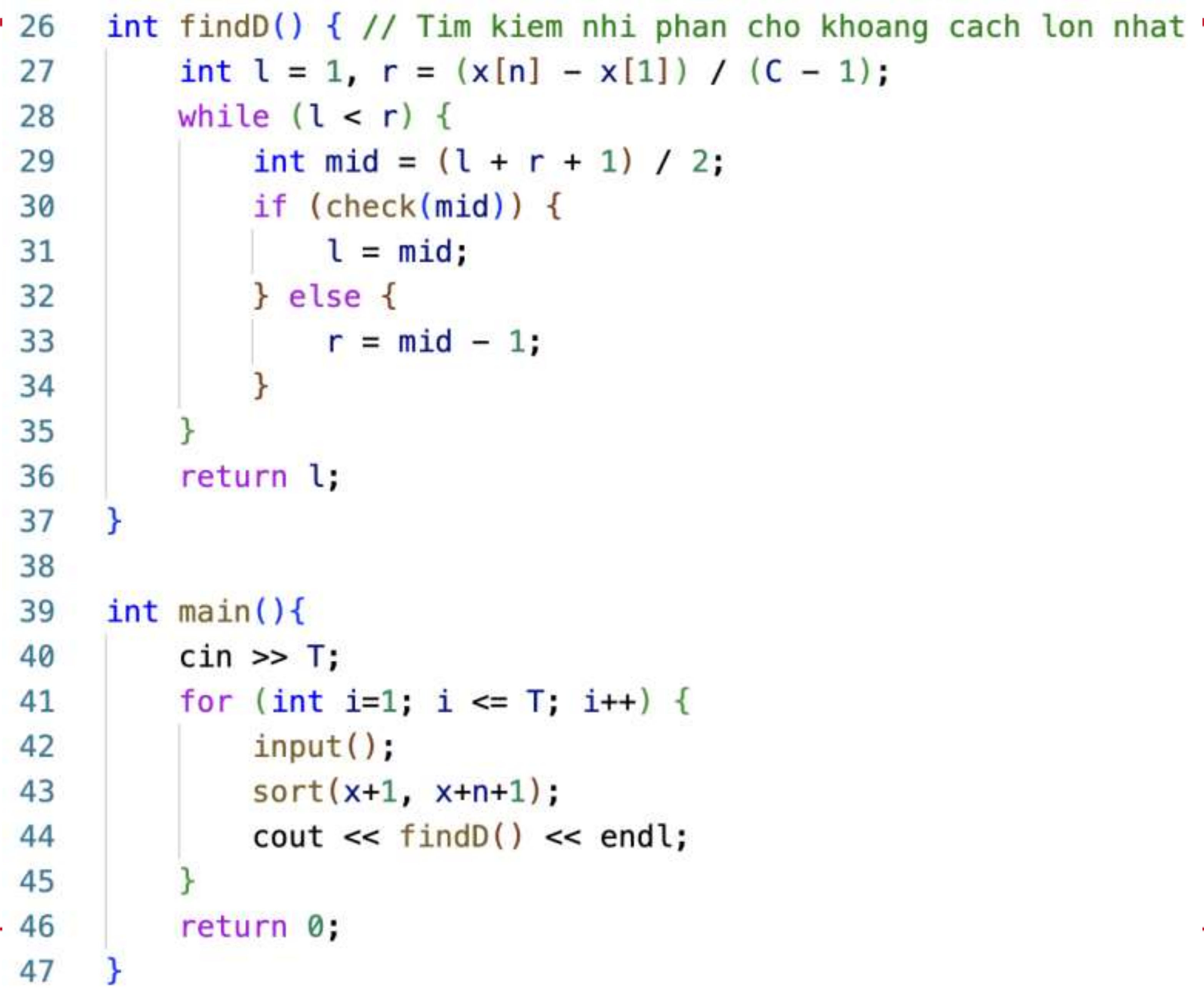
- Idea to solve: Sort $a_1,\ldots, a_N$ in increasing order
- The max-distance must be $\leq \frac{a_N - a_1}{c-1}$
- Find max-distance from $\mathrm{d} = \frac{a_N - a_1}{c-1}$ down to 1: this can be improved
  - Binary search for d
  - Complexity: O(n log n)

Max-distance sub-sequence - Implementation



```cpp
#include <bits/stdc++.h>
using namespace std;
#define N 100001
int n, C, T, x[N];
void input() {
    cin >> n >> C;
    for (int i = 1; i <= n; i++) {
        cin >> x[i];
    }
}
bool check(int d) {
    // Kiem tra xem co the lay tap con C voi khoang cach d
    int nb = 1;
    int last = x[1];
    for (int i = 2; i <= n; i++) {
        if (x[i] - last >= d) {
            last = x[i];
            nb++;
            if (nb == C)
                break;
        }
    }
    return nb>=C;
}
```

Max-distance sub-sequence - Implementation



```
int findD() { // Tim kiem nhi phan cho khoang cach lon nhat
    int l = 1, r = (x[n] - x[1]) / (C - 1);
    while (l < r) {
        int mid = (l + r + 1) / 2;
        if (check(mid)) {
            l = mid;
        } else {
            r = mid - 1;
        }
    }
    return l;
}

int main(){
    cin >> T;
    for (int i=1; i <= T; i++) {
        input();
        sort(x+1, x+n+1);
        cout << findD() << endl;
    }
    return 0;
}
```

THANK YOU !