On Disk Scheduling

To gain optimal head movement distance, observe that:

I. The request cylinder numbers can be thought to be lied on a one-dimensional line.

II. Define a segment by the smallest request number and the largest request number, suppose that the head is in the segment:

   A. We can satisfy all requests without moving the head out of the segment. And if there exists an optimal scheduling that moves the head out of the segment, we can construct a scheduling restricting the head in the segment without increasing the total movement distance.

   B. The head must go to the each end of the segment at least once, and if the head goes from one end to another, it can satisfy all requests along this move.

Therefore, a solution to this problem is:

1. Move the head from the initial position to one end of the segment, which is closer to the initial position.
2. Move the head from this end to another end.

```
#include <cstdio>
#include <iostream>
#include <string>
#include <sstream>
#include <algorithm>
using namespace std;
int main() {
        int cases;
        scanf("%d", &cases); while (getchar() != '\n');
        string s;
        while (cases--) {
                getline(cin, s);
                stringstream ss(s);
                int mid;
                ss >> mid;
                int small = mid, big = mid, d;
                while (ss >> d) {
                        if (d<small) small = d;
                        if (d>big) big = d;
                }
                printf("%d\n", min(mid-small, big-mid)+big-small);
        }
}
```