

과제 #9 카페 - 201921438 조현태

1) 소스코드

```
#include <stdio.h>
#include <iostream>
#include <algorithm>
#include <vector>
#include <stdlib.h>
using namespace std;

// n = 카페의 수, k = 방문할 카페의 수, s = 현재 위치
// 1 <= k <= n <= 1000000
// -10억 <= s <= 10억
int n, k, s;

vector<int> cafe; // 카페의 위치

// 함수 선언
int find_cafe();

int main()
{
    cin >> n >> k >> s;

    for (int i = 0; i < n; i++)
    {
        int a;
        cin >> a;
        cafe.push_back(a);
    }

    // 함수 실행
    cout << find_cafe() << endl;

    return 0;
}

// 정해진 개수의 카페를 모두 가는 최단거리를 구하는 함수
int find_cafe()
{
    // 카페의 위치 오름차순 정렬
    sort(cafe.begin(), cafe.end());
```

```

// 초기 정답 ( 0 ~ k번째 카페라고 가정)
// 첫번째 카페와 현재 위치 사이의 거리
int dist = abs(s - cafe[0]);
// 만약 k번째 카페와 현재 위치 사이의 거리가 더 작다면
if (abs(s - cafe[0]) > abs(s - cafe[k-1]))
    dist = abs(s - cafe[k-1]); // k번째 카페와 현재 위치 사이의 거리로 바꾼다.
// (첫번째 카페부터 k번째 카페까지의 거리 + 배열과 현재 위치 사이의 최단거리)
int answer = abs(cafe[0] - cafe[k-1]) + dist;

// 모든 카페에 대해서 탐색
int new_answer = 0;
for (int i = 1; i < n - k+1; i++)
{
    // i ~ i+k번째 카페라고 가정
    int dist = abs(s - cafe[i]);
    if (abs(s - cafe[i]) > abs(s - cafe[i+k-1]))
        dist = abs(s - cafe[i+k-1]);
    new_answer = abs(cafe[i] - cafe[i+k-1]) + dist;

    // 새로운 정답이 더 작을 경우
    if (answer > new_answer)
        answer = new_answer;
}
return answer;
}

```

2) 문제 설명

카페가 1 ~ 100만까지의 x좌표 상에 있고
내 위치(s)는 -10억 ~ 10억까지의 x 좌표 상에 있다고 할 때,
원하는 수(k)만큼의 카페에 도달할 때까지의 최소거리를 구하는 문제입니다.

처음에는 시작점에서 가까운 거리의 점까지 이동하는 그리드를 사용하려고 했지만
x좌표에 있기 때문에 이동할때마다 점의 거리가 바뀌는 문제가 있었습니다.
그래서 차리리 k개의 카페를 미리 만들어서 판단하는 것이 좋겠다고 생각합니다.

k개의 카페 배열을 만들고 카페의 크기를 구합니다.
이후, 현재 위치에서 임의의 카페 배열의 첫 번째 카페와 마지막 카페의 거리 중
더 가까운 거리 구해서 카페 크기에 더해주면 됩니다.
그리고 첫 번째 카페를 제거하고 다음 위치의 카페를 추가해서 계속 계산을 반복합니다.
그때마다 계산되는 정답을 계속 최솟값을 최신화해주면 정답이 됩니다.

ex)
 $n = 6, k = 3, s = 6$
 $cafe = [1, 2, 5, 7, 7, 9]$

1) [1, 2, 5]일 때
배열의 크기 = 4
첫 번째 카페에서 현재 위치까지의 거리 = 5
마지막 카페에서 현재 위치까지의 거리 = 1
 $\min(5, 1) = 1$
 $\therefore 4 + 1 = 5$

2) [2, 5, 7]일 때
배열의 크기 = 5
 $\min(4, 1) = 1$
 $\therefore 5 + 1 = 6$

3) [5, 7, 7]일 때
배열의 크기 = 2
 $\min(1, 1) = 1$
 $\therefore 2 + 1 = 3$

4) [7, 7, 9]일 때
배열의 크기 = 2
 $\min(1, 3) = 1$
 $\therefore 2 + 1 = 3$

따라서 정답은 3입니다.

하지만 n, k의 크기가 100만이고 배열의 첫 번째 값과 마지막 값만 이용하기 때문에
배열을 사용하는 것은 매우 큰 낭비라고 생각하고 실제로도 극단적인 테스트 케이스에서는
시간 초과, 오류가 발생하는 것을 확인했습니다.
따라서 위와 같이 배열을 사용한다고 가정하고 식을 간소화하여
for문을 이용하여 계산하였습니다.