##### 공항 건설하기

Level 4

1보다 큰 N개의 도시 중 한 곳에 공항을 지을 예정입니다. 사람들의 편의를 위해 공항으로부터 각 사람들까지의 도시간 이동 거리가 최소가 되는 도시에 짓기로 하였습니다. 편의상 도시는 일직선상에 놓여있다고 가정하며 **좌표의 범위는 음수가 포함**됩니다. 또한 좌표는 정렬되어 있지 않습니다. 직선상의 위치와 그 도시에 사는 사람들의 수가 주어질 때, 공항을 지을 도시의 위치를 반환해주는 함수 chooseCity 함수를 완성하세요. 거리가 같은 도시가 2개 이상일 경우 위치가 더 작은 쪽의 도시를 선택하면 됩니다. 예를 들어 다음과 같은 정보의 도시가 있다고 가정해 봅시다.

| **위치** | **1** | **2** | **3** |
| --- | --- | --- | --- |
| 인구수 | 5 | 2 | 3 |

이 살 경우, 각각의 도시에 공항을 지었을 때의 사람들의 이동 거리는 **8, 8, 12** 이므로 1번 또는 2번에 지을 수 있지만, **1의 위치가 더 작으므로** 1을 반환해주면 됩니다.

#include "stdafx.h"

#include<vector>

#include<iostream>

#include<algorithm>

using namespace std;

int distance(pair<int, long long> a, pair<int, long long> b)

{

if (a.first > b.first)

return (a.first - b.first) \* b.second;

else

return (b.first - a.first) \* b.second;

}

int chooseCity(int n, vector<pair<int, int>> city)

{

int answer = 0;

vector<pair<int, long long>> dis;

sort(city.begin(), city.end());

int mid = city.size() / 2;

for (int i = -1; i <= 1; i++)

{

pair<long int, long long> tmp;

tmp.first = city[mid - i].first;

tmp.second = 0;

for (int j = 0; j < city.size(); j++)

{

if (i == j) continue;

else

tmp.second += distance(city[mid - i], city[mid - i]);

}

dis.push\_back(tmp);

}

for (int i = 1; i < dis.size(); i++)

{

if (dis[0].second > dis[i].second)

dis[0] = dis[i];

else if (dis[0].second == dis[i].second)

dis[0] = ((dis[0].first < dis[i].first) ? dis[0] : dis[i]);

else

continue;

}

answer = dis[0].first;

return answer;

}

int main()

{

int tn = 3;

pair<int, int> a, b, c;

a.first = 1, a.second = 5;

b.first = 2, b.second = 2;

c.first = 3, c.second = 3;

vector<pair<int, int>> tcity{ a,b,c };

//아래는 테스트 출력을 위한 코드입니다.

cout << chooseCity(tn, tcity);

return 0;

}

/\*

과정 설명

\*/