문제 설명

**캐시**

지도개발팀에서 근무하는 제이지는 지도에서 도시 이름을 검색하면 해당 도시와 관련된 맛집 게시물들을 데이터베이스에서 읽어 보여주는 서비스를 개발하고 있다.  
이 프로그램의 테스팅 업무를 담당하고 있는 어피치는 서비스를 오픈하기 전 각 로직에 대한 성능 측정을 수행하였는데, 제이지가 작성한 부분 중 데이터베이스에서 게시물을 가져오는 부분의 실행시간이 너무 오래 걸린다는 것을 알게 되었다.  
어피치는 제이지에게 해당 로직을 개선하라고 닦달하기 시작하였고, 제이지는 DB 캐시를 적용하여 성능 개선을 시도하고 있지만 캐시 크기를 얼마로 해야 효율적인지 몰라 난감한 상황이다.

어피치에게 시달리는 제이지를 도와, DB 캐시를 적용할 때 캐시 크기에 따른 실행시간 측정 프로그램을 작성하시오.

**입력 형식**

* 캐시 크기(cacheSize)와 도시이름 배열(cities)을 입력받는다.
* cacheSize는 정수이며, 범위는 0 ≦ cacheSize ≦ 30 이다.
* cities는 도시 이름으로 이뤄진 문자열 배열로, 최대 도시 수는 100,000개이다.
* 각 도시 이름은 공백, 숫자, 특수문자 등이 없는 영문자로 구성되며, 대소문자 구분을 하지 않는다. 도시 이름은 최대 20자로 이루어져 있다.

**출력 형식**

* 입력된 도시이름 배열을 순서대로 처리할 때, "총 실행시간"을 출력한다.

**조건**

* 캐시 교체 알고리즘은 LRU(Least Recently Used)를 사용한다.
* cache hit일 경우 실행시간은 1이다.
* cache miss일 경우 실행시간은 5이다.

**입출력 예제**

| **캐시크기(cacheSize)** | **도시이름(cities)** | **실행시간** |
| --- | --- | --- |
| 3 | ["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "Jeju", "Pangyo", "Seoul", "NewYork", "LA"] | 50 |
| 3 | ["Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul"] | 21 |
| 2 | ["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "SanFrancisco", "Seoul", "Rome", "Paris", "Jeju", "NewYork", "Rome"] | 60 |
| 5 | ["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "SanFrancisco", "Seoul", "Rome", "Paris", "Jeju", "NewYork", "Rome"] | 52 |
| 2 | ["Jeju", "Pangyo", "NewYork", "newyork"] | 16 |
| 0 | ["Jeju", "Pangyo", "Seoul", "NewYork", "LA"] | 25 |

#include <iostream>

#include <string>

#include <vector>

using namespace std;

bool compare(int n, vector<string> cities, vector<string> cache)

{

for (int i = 0; i < cache.size(); i++)

{

if (cities[n] == cache[i])

return true;

}

return false;

}

void lru(vector<string> &cache, string str)

{

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

{

cache[i - 1] = cache[i];

}

cache[cache.size() - 1] = str;

}

int solution(int cacheSize, vector<string> cities) {

int answer = 0;

int hit = 1;

int miss = 5;

vector<string> cache(cacheSize, "");

if (cacheSize == 0)

return answer = cities.size() \* miss;

for (int i = 0; i < cities.size(); i++)

{

if (compare(i, cities, cache))

{

answer += hit;

}

else

{

lru(cache, cities[i]);

answer += miss;

}

}

return answer;

}

int main()

{

// 50

int cacheSize = 3;

vector<string> cities = { {"Jeju"}, {"Pangyo"}, {"Seoul"}, {"NewYork"}, {"LA"}, {"Jeju"}, {"Pangyo"}, {"Seoul"}, {"NewYork"}, {"LA"} };

// 21

//cacheSize = 3;

//cities = {{"Jeju"}, {"Pangyo"}, {"Seoul"}, {"Jeju"}, {"Pangyo"}, {"Seoul"}, {"Jeju"}, {"Pangyo"}, {"Seoul"}};

//60

//cacheSize = 2;

//cities = { {"Jeju"}, {"Pangyo"}, {"Seoul"}, {"NewYork"}, {"LA"}, {"SanFrancisco"}, {"Seoul"}, {"Rome"},

// {"Paris"}, {"Jeju"}, {"NewYork"}, {"Rome"} };

// 52

cacheSize = 5;

cities = { {"Jeju"}, {"Pangyo"}, {"Seoul"}, {"NewYork"}, {"LA"}, {"SanFrancisco"}, {"Seoul"},

{"Rome"}, {"Paris"}, {"Jeju"}, {"NewYork"}, {"Rome"} };

int a = solution(cacheSize, cities);

cout << a << endl;

return 0;

}