

1. 주유소

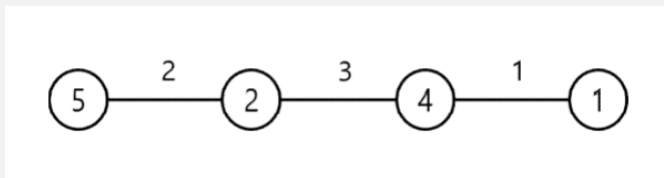
문제 파악

1. 도시는 순서대로 방문, 경로는 하나로 고정되어 있음
2. 도시마다 기름값과 경로의 길이가 주어짐
3. 기름은 무제한으로 넣을 수 있고, 경로 1(km)당 기름 1(원)이 소모됨
4. 최대한 적은 비용으로 처음부터 마지막 도시까지 이동

1. 주유소

문제 풀이

그리디한 접근법으로 최대한 싼 기름값 유지하면서 이동한다.



1. 거리 2를 이동해서 도시2로 이동, 도시2의 기름값이 더 싸므로 거리 2만큼만 도시1에서 주유
2. 거리 3을 이동해서 도시3으로 이동, 도시3의 기름값이 더 비싸므로 도시3에서 추가 주유할 이유가 없음
3. 거리 1을 더 이동해서 목적지(도시4)에 도착, 도시2에서 거리4(=3+1)만큼 주유

총 비용 $5*2+4*2=18$

1. 주유소

핵심 코드

```
ll cur_price = price[0], cur_len = 0;
for (ll i = 0; i < n - 1; i++) {
    cur_len += len[i];
    if (i == n - 2) { // 맨 마지막은 그냥 계산
        tot += cur_price * cur_len;
    }
    else if (cur_price > price[i + 1]) { // 현재 도시보다 다음 도시의 가격이 싸질때까지 누적
        tot += cur_price * cur_len;
        cur_price = price[i + 1];
        cur_len = 0;
    }
}
```

마지막 도시의 기름값은 고려할 필요 없으므로 무시한다

2. 문자열 게임 2

문제 파악

1. 알파벳 소문자 문자열 w , 자연수 k 가 주어지고
2. 어떤 문자를 정확하게 k 개 포함하는 가장 짧은 연속 문자열 길이
3. 어떤 문자를 정확하게 k 개 포함하는 가장 긴 연속 문자열 길이
4. 총 t 회 반복

2. 문자열 게임 2

문제 파악

어떤 문자 k개 => 모든 알파벳 소문자에 대해 검사

어떤 문자를 정확하게 k개 포함하는 가장 짧은 연속 문자열

=어떤 문자가 정확하게 k개 포함되어 있고 양끝 문자열이 그 문자이면서
그 간격이 제일 짧은 것

가장 긴 연속 문자열도 마찬가지로

2. 문자열 게임 2

핵심 코드

```
// 어떤 문자를 정확하게 k개 포함하는 최소/최대 길이 문자열은
// 양끝의 문자 간격이 최소/최대가 된다
int minsub = w.size() + 1, maxsub = -1; // 문자열 길이
int min_st, min_end, max_st, max_end; // 문자열 양끝 위치
for (int j = 0; j < 26; j++) {
    if (al[j][0] < k) continue;
    for (int l = 1; l + k - 1 < al[j].size(); l++) {
        if (minsub > al[j][l + k - 1] - al[j][l]) { // 최소 간격
            minsub = al[j][l + k - 1] - al[j][l];
            min_st = al[j][l]; min_end = al[j][l + k - 1];
        }
        if (maxsub < al[j][l + k - 1] - al[j][l]) { // 최대 간격
            maxsub = al[j][l + k - 1] - al[j][l];
            max_st = al[j][l]; max_end = al[j][l + k - 1];
        }
    }
}
```