

-1217-

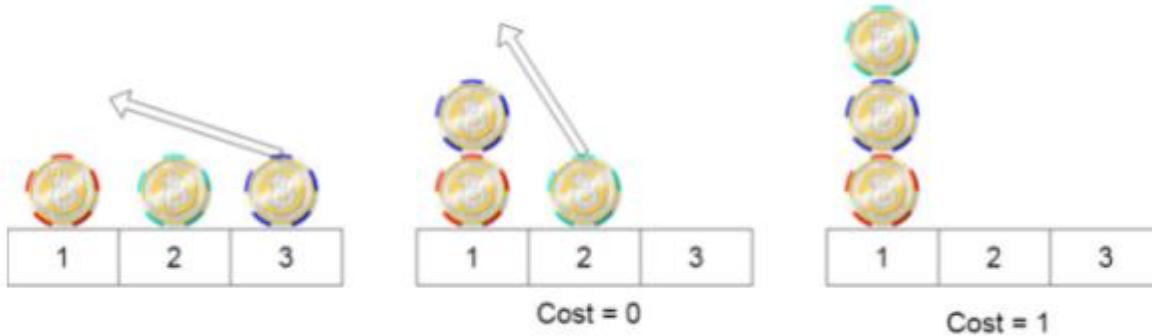
# Minimum Cost to Move Chips to The Same Position

202021006 소프트웨어학과 김예진

# 1217 – Minimum Cost to Move Chips to The Same Position

그림과 같이 각 position에 chip들이 있을 때, 각 chip들을 한 position으로 모으기 위한 최소 cost를 구하는 문제

Example 1:



Input: position = [1,2,3]

Output: 1

Explanation: First step: Move the chip at position 3 to position 1 with cost = 0.

Second step: Move the chip at position 2 to position 1 with cost = 1.

Total cost is 1.

Ex)

3번 -> 1번

- cost가 아예 들지 않음.

2번 -> 1번

- 1cost

따라서 최소 cost 는 1

- 좌 우 상관없이 2칸을 이동할때는 cost = 0
- 1칸 이동시 cost = 1

# 접근 방법

- 1) 문제를 보면 어차피 짝수 인덱스에 있는 칩은 비용이 필요하지 않음
- 2) 짝수 인덱스에 있는 칩과 홀수 인덱스에 있는 칩의 수를 계산해서 더 적은 쪽을 옮겨주면 됨
- 3) 따라서, position 벡터를 돌면서 짝수, 홀수의 개수를 체크해준 다음 더 적은 쪽을 반환.

# 코드 설명

```
class Solution {  
public:
```

```
    int minCostToMoveChips(vector<int>& position) {  
        int even = 0;  
        int odd = 0;
```

-> even : 짝수 인덱스에 있는 chip의 개수  
-> odd : 홀수 인덱스에 있는 chip의 개수

```
        int len = position.size();
```

```
        for (int i = 0; i < len; i++){  
            if (position[i] % 2 == 0)  
                even += 1;  
            else  
                odd += 1;  
        }
```

-> position 벡터의 길이만큼 for문을 돌면서 짝수, 홀수의 개수를 각각 구해준다.

```
        return min(even, odd);
```

-> 둘 중 더 작은 값을 반환한다.

```
    }
```

```
};
```