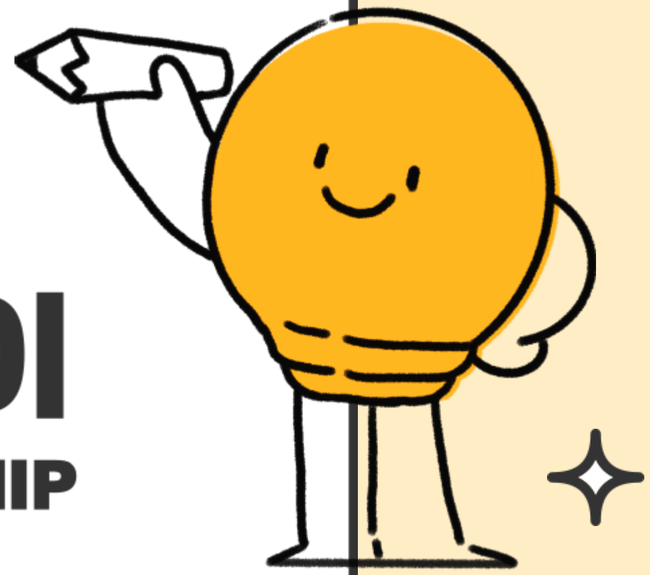
2024년도 상반기 널카말카 세미나

알고리즘 문제풀이

2024 KAKAO WINTER INTERNSHIP

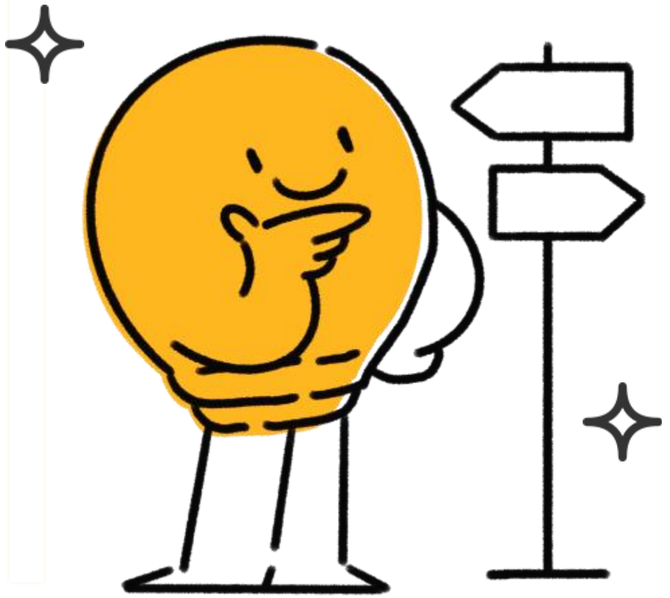
주사위 고르기



남윤서

목차

오늘의 문제?



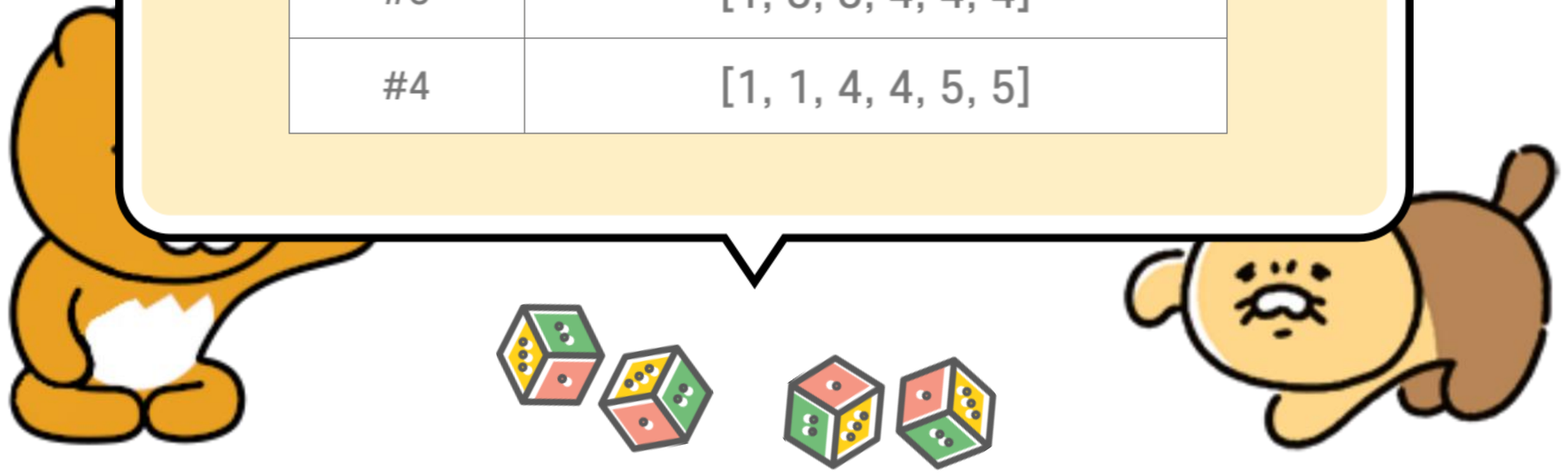
주사위 고르기

A와 B가 n개의 주사위를 가지고 승부를 합니다. 주사위의 6개 면에 각각 하나의 수가 쓰여 있으며, 주사위를 던졌을 때 각 면이 나올 확률은 동일합니다. 각 주사위는 1 ~ n의 번호를 가지고 있으며, 주사위에 쓰인 수의 구성은 모두 다릅니다.

A가 먼저 n / 2개의 주사위를 가져가면 B가 남은 n / 2개의 주사위를 가져갑니다. 각각 가져간 주사위를 모두 굴린 뒤, **나온 수들을 모두 합해 점수를 계산**합니다. **점수가 더 큰 쪽이 승리하며, 점수가 같다면 무승부입니다.**

A는 자신이 승리할 확률이 가장 높아지도록 주사위를 가져가려 합니다.

주사위	구성
#1	[1, 2, 3, 4, 5, 6]
#2	[3, 3, 3, 3, 4, 4]
#3	[1, 3, 3, 4, 4, 4]
#4	[1, 1, 4, 4, 5, 5]



라이언이 가져가는 주사위 조합에 따라, 주사위를 굴린 1296가지 경우의 승패 결과

라이언의 주사위	승	무	패
#1 , #2	596	196	504
#1 , #3	560	176	560
#1 , #4	616	184	496
#2 , #3	496	184	616
#2 , #4	560	176	560
#3 , #4	504	196	596

라이언이 가져가는 주사위 조합에 따라, 주사위를 굴린 1296가지 경우의 승패 결과

라이언의 주사위	승	무	패
#1 , #2	596	196	504
#1 , #3	560	176	560
#1 , #4	616	184	496
#2 , #3	496	184	616
#2 , #4	560	176	560
#3 , #4	504	196	596

우선 어떻게 코드를 구성할지 생각해봅시다.

A가 가져갈 주사위를 선택하는 부분

[#1,#2] [#1,#3]
[#1,#4] [#1,#5]
⋮
[#8,#9] [#9,#10]

+

가져간 주사위를 굴린 결과를 세는 부분

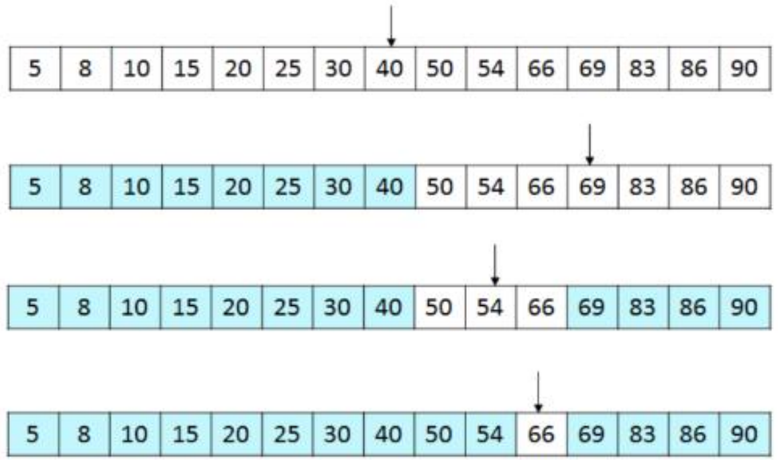
[#1,#2] 승
[#1,#3] 패
[#1,#4] 무
⋮

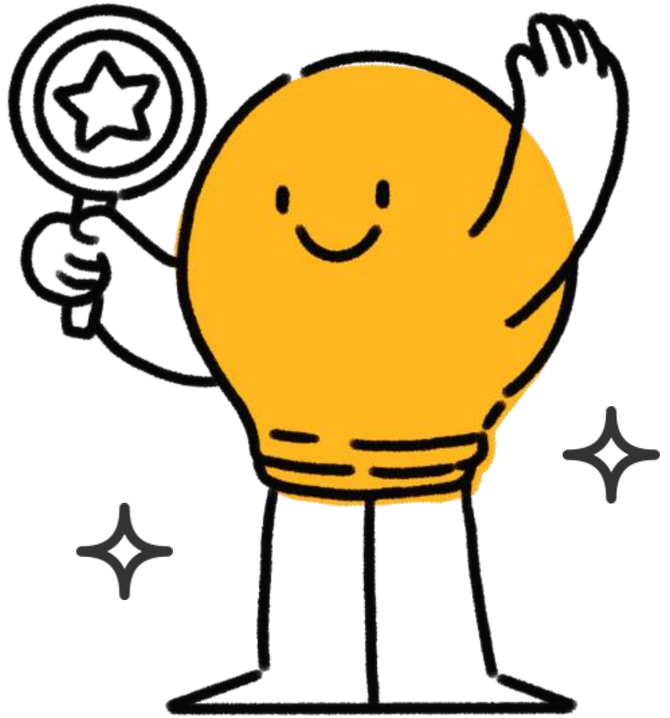
문제를 풀어봅시다!

1	2	3
완전 탐색	시간 초과	이분탐색

생각해야할 것들은 위와 같습니다.

이분탐색(이진탐색)이란?


5 8 10 15 20 25 30 40 50 54 66 69 83 86 90
5 8 10 15 20 25 30 40 50 54 66 69 83 86 90
5 8 10 15 20 25 30 40 50 54 66 69 83 86 90
5 8 10 15 20 25 30 40 50 54 66 69 83 86 90



문제 풀이 코드 작성

주사위 고르기

```python
from itertools import combinations

def solution(dice):
    dic = {} #주사위의 구성
    L = len(dice) #주사위의 개수

    for A_combi in combinations(range(L), L//2):  #combination를 활용하여 주사위 개수와 2분의 L개를 각각 굴려 A 조합 구성
        B_combi = []
        for i in range(L):
            if i not in A_combi:   #A의 주사위가 아닌 나머지 주사위를 B_combi에 넣음.
                B_combi.append(i)
```

문제 풀이 코드 작성

만들 수 있는 주사위의 합 구하기

```
        A = []  #A 리스트를 만듦.
        for order_product in product(range(6), repeat=L // 2):  #product를 이용해 order_product에 만들 수 있는 주사위 면들의 조합을 만듦.
            sum_A = 0
            for i, j in zip(A_combi, order_product):     #A_combi(굴릴 주사위의 번호 조합), order_product(만들 수 있는 주사위면 조합) 패킹함.
                sum_A += dice[i][j] #각 주사위별로 만들 수 있는 조합의 합을 구함.
            A.append(sum_A)

        B = []  #B 리스트를 만듦.
        for order_product in product(range(6), repeat=L // 2):
            sum_B = 0
            for i, j in zip(B_combi, order_product):
                sum_B += dice[i][j]
            B.append(sum_B)
        B.sort()    #B 리스트는 이후에 있을 이분탐색을 위해 정렬함.
```

문제 풀이 코드 작성

이분탐색(이진탐색)

```
29          wins = 0
30          for num in A:   #A에서 num을 가져오고
31              wins += bisect_left(B, num) #이진탐색을 해줌.
32                                          #A의 주사위 합과 B의 주사위 합을 각각 비교해 A가 더 클 경우에 wins에 넣어줌.
33          dic[wins] = list(A_combi) #A_combi(주사위 조합)에서 wins값을 key로하여 딕셔너리에 저장함.
34
```

문제 풀이 코드 작성

이분탐색(이진탐색)

```
29          wins = 0
30          for num in
31              wins +
32
33          dic[wins]
34
```

```
#                   5                               A의 원소
[0, 1, 2, 3, 4, 5, 5, 5, 5, 6, 7, 8, 9,] #B리스트


#                                  7                A의 원소
[0, 1, 2, 3, 4, 5, 5, 5, 5, 6, 7, 8, 9,] #B리스트
```

클 경우에 wins에 넣어줌.
저장함.

문제 풀이 코드 작성

이분탐색(이진탐색)

```
29          wins = 0
30          for num in A:   #A에서 num을 가져오고
31              wins += bisect_left(B, num) #이진탐색을 해줌.
32                                          #A의 주사위 합과 B의 주사위 합을 각각 비교해 A가 더 클 경우에 wins에 넣어줌.
33          dic[wins] = list(A_combi) #A_combi(주사위 조합)에서 wins값을 key로하여 딕셔너리에 저장함.
34
```

문제 풀이 코드 작성

결과 반환

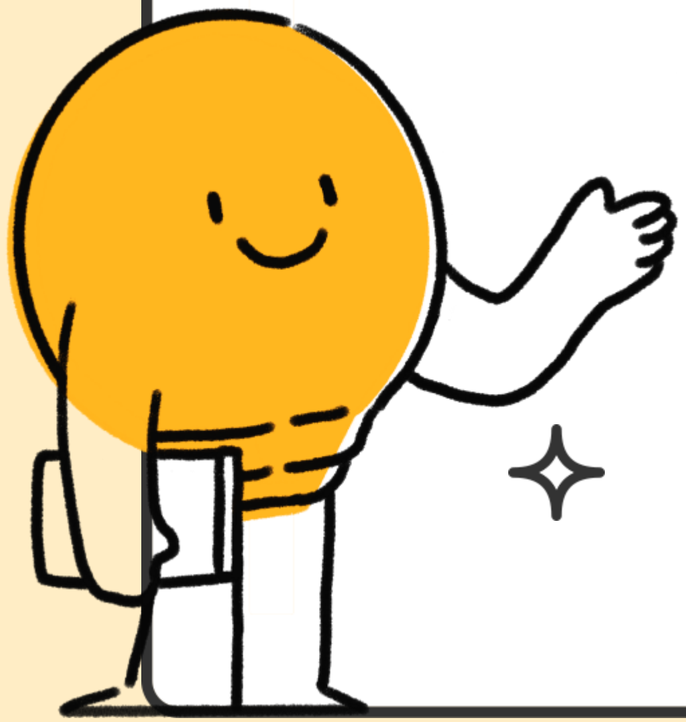
```
35      max_key = max(dic.keys())   #가장 큰 key값을 max_key에 넣어줌.
36
37      max_value = dic[max_key]    #제일 큰 값을 max_value에 넣어줌.
38      result = []
39      for x in max_value: #그리고 아까 인덱스값을 넣었기 때문에 0부터 시작함.
40          result.append(x + 1)    #따라서 1씩 더해줌.
41
42      return result
```

총 코드

solution.py

```python
from itertools import combinations, product
from bisect import bisect_left

def solution(dice):
    dic = {}    #주사위의 구성
    L = len(dice)   #주사위의 개수

    for A_combi in combinations(range(L), L // 2):  #combination를 활용하여 주사위 개수와 2분의 L개를 각각 골라 A 조합 구성
        B_combi = []
        for i in range(L):
            if i not in A_combi:    #A의 주사위가 아닌 나머지 주사위를 B_combi에 넣음.
                B_combi.append(i)

        A = []  #A 리스트를 만듦.
        for order_product in product(range(6), repeat=L // 2):  #product를 이용해 order_product에 만들 수 있는 주사위 면들의 조합을 만듦.
            sum_A = 0
            for i, j in zip(A_combi, order_product):     #A_combi(골랐 주사위의 번호 조합), order_product(만들 수 있는 주사위면 조합) 페어링함.
                sum_A += dice[i][j] #각 주사위별로 만들 수 있는 조합의 합을 구함.
            A.append(sum_A)

        B = []  #B 리스트를 만듦.
        for order_product in product(range(6), repeat=L // 2):
            sum_B = 0
            for i, j in zip(B_combi, order_product):
                sum_B += dice[i][j]
            B.append(sum_B)
        B.sort()    #B 리스트는 이후에 있을 이분탐색을 위해 정렬함.

        wins = 0
        for num in A:   #A에서 num를 가져오고
            wins += bisect_left(B, num) #이진탐색을 해줌. A의 주사위 합과 B의 주사위 합을 각각 비교해 A가 더 클 경우에 wins에 넣어줌.
        dic[wins] = list(A_combi) #A_combi(주사위 조합)에서 wins값을 key로하여 딕셔너리에 저장함.


    max_key = max(dic.keys())   #가장 큰 key값을 max_key에 넣어줌.

    max_value = dic[max_key]    #제일 큰 값을 max_value에 넣어줌.
    result = []
    for x in max_value: #그리고 아까 인덱스값을 넣었기 때문에 0부터 시작함.
        result.append(x + 1)    #따라서 1씩 더해줌.

    return result
```

마우리



감사합니다