

Report

프로그래머스 폰켓몬 문제(1845)



과목 : 코딩테스트지도

담당교수 : 이승진 교수님

학부 : IT융합자율학부

학번 : 202114136

이름 : 장준희

제출일 : 2025-03-14

조건

폰켓몬의 $n/2$ 가지만 가질 수 있음

가장 많은 종류의 폰켓몬을 고르려고 함

그 때의 폰켓몬 종류 번호의 개수를 리턴하기

제한사항

nums는 폰켓몬의 종류 번호가 담긴 1차원 배열입니다.

nums의 길이(N)는 1 이상 10,000 이하의 자연수이며, 항상 짝수로 주어집니다.

폰켓몬의 종류 번호는 1 이상 200,000 이하의 자연수로 나타냅니다.

가장 많은 종류의 폰켓몬을 선택하는 방법이 여러 가지인 경우에도, 선택할 수 있는 폰켓몬 종류 개수의 최댓값 하나만 return 하면 됩니다. (출처: 프로그래머스)

1차 시도

폰켓몬의 종류를 최대한 많이 가지는 것이 목표이다.

그러면 Set을 이용하여 같은 번호를 제거한 후 크기를 재면 간단하게 폰켓몬 종류 번호의 개수를 구할 수 있을 것이다.

먼저 모든 폰켓몬중에 모든 경우의 수를 구하는 로직을 구현해 보자

(괄호 안은 인덱스(0으로 시작) 기준으로 작성됨) (여기에서 n은 폰켓몬 배열 크기)

첫번째 번호는 처음부터(0) $n/2$ 자리까지 ($n/2-1$)

두번째 번호는 첫번째의 다음 인덱스($n+1$)부터 마지막 바로 전까지 ($nums.length-2$)

세번째 번호는 두번째의 다음 인덱스($n+1$)부터 마지막 까지 ($nums.length-1$)

위의 경우는 폰켓몬이 6개일때만 해당된다.

이를 바탕으로 모든 폰켓몬 개수에 적용이 가능하도록 해보자

last 변수를 두어 초기값을 $\text{nums.length}/2$ 로 둔다.

그리고 두번째 부터는 이전의 index의 다음부터 last값을 1씩 늘려가며 last가 $n-1$ 이 될 때까지 반복한다.

이를 반복하면 모든 폰켓몬의 경우의 수를 구할 수 있을 것이다.

하지만 이렇게 구현하려고 하니 for문은 어떻게 구현해야 되는지부터 난관에 빠지게 된다.

또한 $n/2$ 개를 선택하는 모든 조합을 고려하는 방식은 **비효율적**이고 구현도 복잡했음.

2차 시도

위의 방법(모든 경우의 수를 계산하는 방법)은 비효율 적이다.

이 문제를 더 간단하게 생각해 보도록 하겠다.

폰켓몬중 $n/2$ 만을 선택한다.

최대한 많은 폰켓몬 종류를 고르려 한다.

그때 폰켓몬 "종류 번호의 개수"만 리턴하면 되므로 다음과 같이 조건을 만들 수 있다.

폰켓몬중 $n/2$ 를 골라야 하므로 $n/2$

최대한 많은 폰켓몬을 고르기 위해서는 같은 종류는 1개만 남기기 위해서 Set을 이용한다.

Set의 크기를 측정하면 선택할 수 있는 폰켓몬 종류 개수의 최댓값이 나온다.

하지만 $n/2$ 만 선택할 수 있으므로 Set의 크기보다 $n/2$ 가 작다면 $n/2$ 를 최댓값으로 삼으면 되는 것이다.

Java로 구현하기

폰켓몬중 $n/2$ 를 구할 때에는 폰켓몬을 담은 배열인 nums의 배열 크기를 구하면 된다. (nums는 이미 폰켓몬의 수량에 맞춰서 제공되기 때문)

코드는 `int max = nums.length/2;`

최대한 많은 폰켓몬 종류를 고르려면 배열 값을 set으로 추가 한 후 크기를 구하면 된다.

```
for (int num : nums) {  
    pokemonset.add(num);  
}
```

크기는 이렇게 구한다. `pokemonset.size()`

이중에 작은 값을 구하면 된다. (Math.min 사용)

결과적으로 성공하였다.



```
채점 결과  
정확성: 100.0  
합계: 100.0 / 100.0
```

결과 및 배운 점

이 문제를 풀면서 반드시 복잡한 알고리즘을 사용할 필요가 없다는 것을 깨달았다.

처음에는 모든 경우의 수를 고려해야 한다고 생각했지만, 핵심 개념을 파악하면 훨씬 간단한 방법으로 해결할 수 있었다.

앞으로는 문제를 접할 때 먼저 단순하게 접근해 보고, 가장 효율적인 해결 방법을 찾으려 하겠다.