14888번: 연산자 끼워넣기

연산자 끼워넣기 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	512 MB	596	319	234	51.542%

문제

N개의 수로 이루어진 수열 A_1, A_2, ..., A_N이 주어진다. 또, 수와 수 사이에 끼워넣을 수 있는 N-1개의 연산자가 주어진다. 연산자는 덧셈(+), 뺄셈(-), 곱셈(×), 나눗셈(÷)로만 이루어져 있다.

우리는 수와 수 사이에 연산자를 하나씩 넣어서, 수식을 하나 만들 수 있다. 이 때, 주어진 수의 순서를 바꾸면 안된다.

예를 들어, 6개의 수로 이루어진 수열이 1, 2, 3, 4, 5, 6이고, 주어진 연산자가 덧셈(+) 2개, 뺄셈(-) 1개, 곱셈(×) 1개, 나눗셈(÷) 1개인 경우에는 총 60가지의 식을 만들 수 있다. 예를 들어, 아래와 같은 식을 만들 수 있다.

- 1+2+3-4×5÷6
- 1÷2+3+4-5×6
- 1+2÷3×4-5+6
- 1÷2×3-4-5+6

식의 계산은 연산자 우선순위를 무시하고 앞에서부터 진행해야 한다. 또, 나눗셈은 정수 나눗셈으로 몫만 취한다. 음수를 양수로 나눌 때는 C의 기준을 따른다. 즉 양수로 바꾼 뒤 몫을 취하고, 그 몫을 음수로 바꾼 것과 같다. 이에 따라서, 위의 식 4개의 결과를 계산해보면 아래와 같다.

- 1+2+3-4×5÷6 = 1
- 1÷2+3+4-5×6 = 12
- 1+2÷3×4-5+6 = 5
- 1÷2×3-4-5+6 = -3

N개의 수와 N-1개의 연산자가 주어졌을 때, 만들 수 있는 식의 결과가 최대인 것과 최소인 것을 구하는 프로그램을 작성하시오.

처음 문제를 읽고 든 생각은 .. 리커시브 였다.

전체 숫자열 중 마지막 숫자와 연산자 하나를 선택하고

나머지 숫자열과 나머지 연산자로 다음 함수를 호출 한다.

문제는 연산자의 갯수에 제한이 있으며, 모든 경우의 수 들을 모두 탐색해야 한다는 것이다.

완전 탐색은 탐색 방식의 복잡도가 중요하다.

내 알고리즘의 경우

<> -> 연산자

( ) - > 숫자

(a)<1>(b)<2>(c)<3>(d)<4>(e)<5>(f) 의 경우

숫자 f와 <5>에 들어갈 연산자를 선택한후 다음 함수를 호출한다.

<5> 에 들어갈 수 있는 연산자는 연산자가 모든 연산자가 있다고 가정하면, 4가지이다.

<4> 는 4가지가 가능하다.

<3> 은 3가지

<2> 는 2가지

<1> 은 나머지 1가지가 가능하다.

문자열이 길어진다면

f(n) = 4^n-4^3^2^1 이 되게 된다.

O(f(n)) = 4^n 이다.

<!-소스코드>

```
import copy
nums = []
op =[]

def getInput():
    global nums
    global op
    n = int(input())
    nums = [int(i) for i in input().split()]
    op = [int(i) for i in input().split()]
```

```
    op = [int(i) for i in input().split()]
def calc(index,op):
    tmpOp = copy.deepcopy(op)
    res = []
    t = nums[index]
    if index == 0:
        res.append(t)
        return res
    else :
        for i in range(4):
            if tmpOp[i] !=0:
                tmpOp[i]-=1
                if i == 0 :
                    for j in calc(index-1, tmpOp):
                        res.append(j+t)
                    tmpOp[i]+=1
                elif i == 1 :
                    for j in calc(index-1, tmpOp):
                        res.append(j-t)
                    tmpOp[i]+=1
                elif i == 2 :
                    for j in calc(index-1, tmpOp):
                        res.append(j*t)
                    tmpOp[i]+=1
                else :
                    for j in calc(index-1, tmpOp):
                        if j<0 :
                            res.append(-(-j//t))
                        else : res.append(j//t)
                    tmpOp[i]+=1
        return res


getInput()
answer = calc(len(nums)-1,op)
print (max(answer))
print (min(answer))
```

입력

첫째 줄에 수의 개수 N(2 ≤ N ≤ 11)가 주어진다. 둘째 줄에는 A_1, A_2, ..., A_N이 주어진다. (1 ≤ A_i ≤ 100) 셋째 줄에는 합이 N-1인 4개의 정수가 주어지는데, 차례대로 덧셈(+)의 개수, 뺄셈(-)의 개수, 곱셈(×)의 개수, 나눗셈(÷)의 개수이다.

출력

첫째 줄에 만들 수 있는 식의 결과의 최대값을, 둘째 줄에는 최소값을 출력한다. 최대값과 최소값은 항상 -10억보다 크거나 같고, 10억보다 작거나 같은 결과가 나오는 입력만 주어진다. 또한, 앞에서 부터 계산했을 때, 중간에 계산되는 식의 결과도 항상 -10억보다 크거나 같고, 10억보다 작거나 같다.

예제 입력

```
2
5 6
0 0 1 0
```

예제 출력

```
30
30
```

예제 입력 2

```
3
3 4 5
1 0 1 0
```

예제 출력 2

```
35
17
```

예제 입력 3

예제 입력 3

```
6
1 2 3 4 5 6
2 1 1 1
```

예제 출력 3

```
54
-24
```

힌트

세 번째 예제의 경우에 다음과 같은 식이 최대값/최소값이 나온다.

- 최대값: 1-2÷3+4+5×6
- 최소값: 1+2+3÷4-5×6

출처

- 문제를 만든 사람: baekjoon