Algorithm Week5

15기 정규세션

TOBIG'S 14기 김상현

Contents

15기 정규세션
TOBIG'S 14기 김상현

15기 정규세션
TOBIG'S 14기 김상현

Unit 01 | 3주차 문제 리뷰

Unit 01 | 3주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 1. 엿팔아요

```python
N,C,F = map(int,input().split()) #N = 엿가락의 개수, C=자를때 비용, F=엿 한 가락의 가격

L_list = [int(input()) for _ in range(N)]
max_lenght = max(L_list)
earn_list = []
for i in range(1,max_lenght+1):
    cost = 0
    for j in L_list:
        n = j//i
        if (j/i)%1 == 0:
            cut_count = n-1
        else:
            cut_count = n

        cost += n*i*F - cut_count*C
    earn_list.append(cost) #길이별 가격을 리스트에 추가

print(max(earn_list)) #가장 큰 값을 출력
```

엿을 자를 수 있는 모든 길이를 한 번씩 탐색한다.

엿을 자르는 횟수
1. 자르는 길이로 정확히 나눠지는 경우
2. 자르는 길이로 정확히 나눠지지 않는 경우

이후(엿조각의 개수*길이*가격)-(자른횟수*비용)들을 비교하여 최대 값을 도출하면 된다.

Unit 01 | 3주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 2. IP 채굴기

```python
def dfs(x, y, number):
    if len(number) == 8: #8자리 숫자가 만들어졌다면
        if number not in result: #result에 없다면
            result.append(number)
        return

    dx = [1, -1, 0, 0] #상하좌우 확인 x
    dy = [0, 0, 1, -1] #상하좌우 확인 y
    for k in range(4):
        ddx = x + dx[k]
        ddy = y + dy[k]

        if 0 <= ddx < 4 and 0 <= ddy < 4: #범위 내에 있다면
            dfs(ddx, ddy, number + matrix[ddx][ddy]) #8글자가 될 때 까지 재귀

#입력
matrix = [list(map(str, input().split())) for _ in range(4)]

result = []
for i in range(4):
    for j in range(4):
        dfs(i, j, matrix[i][j]) #0,0부터 3,3까지 모두 검사

print(len(result))
```

숫자판에서 시작할 수 있는 모든 곳에서
재귀함수를 이용해서 모든 경우에 대해 탐색한다.

문제의 조건인 8자리가 되면 result 리스트에 값을
추가하고 함수를 반환시켜 종료 시킨다.

8자리가 되지 않았으면 반복문을 통해 상하좌우로
움직이며 숫자판을 벗어나는 값이 아닌 경우 숫자를
추가 시키며 함수를 계속 반복한다.

마지막에 result 리스트의 원소 개수를 출력하여
모든 경우의 수를 출력한다.

Unit 01 | 3주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 3. 괴도 예은

```python
from itertools import combinations

dia_num, target_price = map(int,input().split())
dia_list = list(map(int, input().split()))
biggest_num = 0

for dias in combinations(card_list,3):
    temp_sum = sum(dias)
    if biggest_num < temp_sum <= target_price:
        biggest_num = temp_sum

print(biggest_num)
```

다이아몬드 가격 리스트에서 3개를 뽑는 모든 경우에 대해 탐색한다.

Itertools의 combinations를 이용하면 모든 조합의 경우를 반환해 준다.

이를 반복문을 통해 target price에 가장 가까우면서 그 값을 넘지 않는 값을 찾아서 출력한다.

Unit 01 | 3주차 문제 리뷰

15기 정규세션
TOBIG'S 14기 김상현

Problem 3. 괴도 예은

```python
dia_num, target_price = map(int, input().split())
dia_list = list(map(int, input().split()))
biggest_num = 0
dia_len = len(dia_list)

for i in range(dia_len-2):
    for j in range(i+1, dia_len-1):
        for k in range(j+1, dia_len):
            temp_sum = dia_list[i]+dia_list[j]+dia_list[k]
            if biggest_num < temp_sum <= target_price:
                biggest_num = temp_sum
print(biggest_num)
```

다이아몬드 가격 리스트에서 3개를 뽑는 모든 경우에 대해 탐색한다.

Itertools의 combinations를 이용하지 않고 3중 for문을 이용해서 푸는 방법으로 최대값을 찾는 조건은 앞의 풀이와 동일하다.

15기 정규세션
TOBIG'S 14기 김상현

Unit 02 | 동적 계획법

Unit 02 | 동적 계획법

동적 계획법(dynamic programming)
복잡한 문제를 간단한 여러 개의 문제로 나누어 푸는 방법을 말한다.

각 하위 문제의 해결을 계산한 뒤, 그 해결책을 저장하여 후에
같은 하위 문제가 나왔을 경우 그것을 간단하게 해결 할 수 있다.

15기 정규세션
TOBIG'S 14기 김상현

예시 피보나치 수열

$$a_n = a_{\{n-1\}} + a_{\{n-2\}}, a_1 = 1, a_2 = 1$$

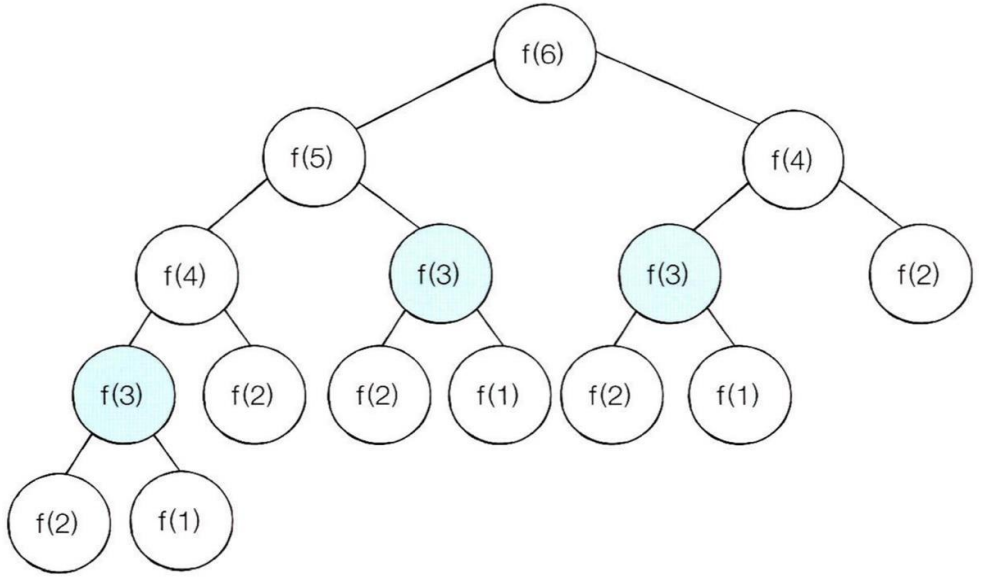
[1,1,2,3,5,8,13,21,34,55,89,...]

15기 정규세션
TOBIG'S 14기 김상현

예시 피보나치 수열

```python
#recursive
def fibo(x):
    if x == 1 or x == 2:
        return 1
    return fibo(x-1) + fibo(x-2)

```



재귀적 방법으로 풀 경우 하위 문제에 대한 값을 계속 다시 구해야 한다.
-> 하위 문제의 값을 기록하자!! (memoization)

15기 정규세션
TOBIG'S 14기 김상현

예시 피보나치 수열

```python
#dynamic programming
def fibo(x):
    if x == 1 or x == 2:
        return 1
    dp = [0]*(x+1)
    dp[1] = 1
    dp[2] = 1

    for i in range(3,x+1):
        dp[i] = dp[i-1]+dp[i-2]

    return dp[x]
```

fibo(x)의 값을 구하기 위해서 fibo(1), fibo(2),...,fibo(x-1)이 필요하므로 이 값들을 차례대로 리스트에 기록하면서 fibo(x)의 값을 구한다.

즉, 하위 문제들의 답을 테이블에 기록해가면서 점화식을 해결해 최종 답을 구한다.

Unit 03 | 5주차 문제 소개

Unit 03 | 5주차 문제 소개

15기 정규세션
TOBIG'S 14기 김상현

Problem 1. 경태는 알고리즘 마스터~

경태는 삼각형 모양의 건물로 되어있는 알고리즘 학원을 등록했다.

이 학원은 매일 아침 꼭대기 층부터 시작하여 방에 출제된 알고리즘 문제를 모두 풀어야 한 층 씩 내려갈 수 있다.

이때 한 층 씩 내려갈 수록 방의 개수는 한 개씩 늘어나며, 각 방마다 출제된 알고리즘 문제의 개수가 다르다. 또한 한 층 내려갈 때에는 현재 위치에서 가장 가까운 왼쪽 대각선 또는 오른쪽 대각선으로만 내려갈 수 있다.

알고리즘 세계 챔피언이 되고 싶은 경태는 매일 최대한 많은 알고리즘 문제를 풀려고 한다. 경태가 그 날 풀 수 있는 알고리즘 문제의 최대 개수를 알아보자!

Unit 03 | 5주차 문제 소개

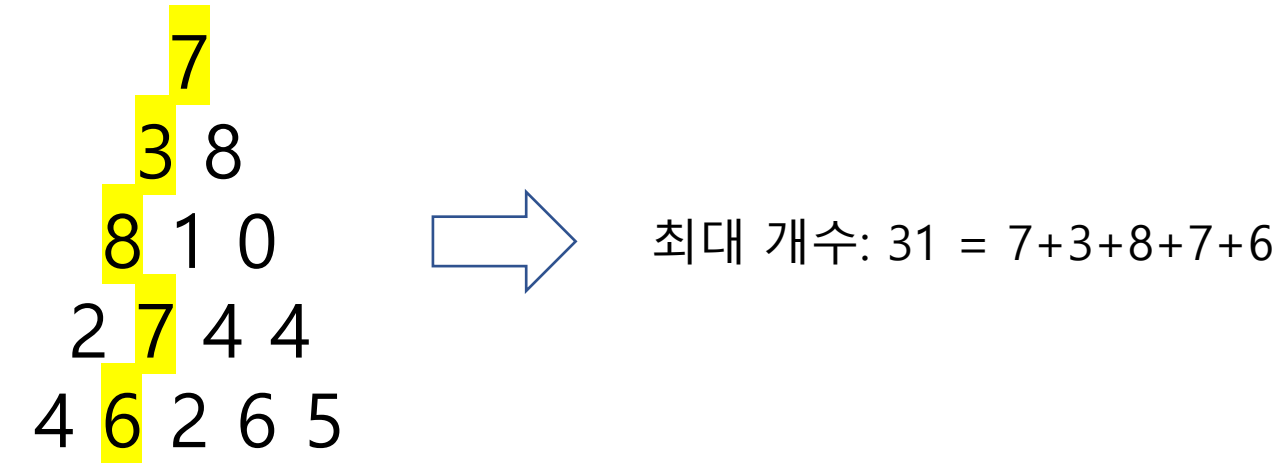
15기 정규세션
TOBIG'S 14기 김상현

Problem 1. 경태는 알고리즘 마스터~

입력으로 알고리즘 건물의 층수 N(1<=N<=20)이 주어지며 그 다음 줄 부터 각 층의 알고리즘 문제 개수가 주어진다.
출력으로 최대 개수를 출력한다.

입력:
5
7
3 8
8 1 0
2 7 4 4
4 6 2 6 5

출력:
31



Unit 03 | 5주차 문제 소개

15기 정규세션
TOBIG'S 14기 김상현

Problem 2. 와인 컨퍼는 즐거워

투빅스 컨퍼런스에서 와인 추천 아이디어를 제시했던 세영이는 와인에 진심이다(궁서체). 컨퍼런스가 끝난 날 세영이는 꿈 속에서 컨퍼런스 팀과 함께 와인바에서 뒷풀이를 하였다. 와인바 사장님께 와인 추천 컨퍼런스 이야기를 하였더니 여러 와인을 무료로 맛볼 수 있는 기회를 주셨다. 하지만 몽땅 마셔버리면 양심이 없으니 티나지 않게 최대한 많이 마실 수 있는 방법을 고민하고 있다.

다양한 와인이 담긴 와인 잔이 일렬로 놓여 있다. 선택한 와인은 잔에 담긴 와인을 모두 마시고, 원래 위치에 놓아야 한다. 또한 연속으로 놓여 있는 3잔을 모두 마실 수는 없다.

N개의 와인 잔에 들어 있는 와인의 양이 주어졌을 때, 가장 많이 마실 수 있는 와인의 양은 얼마인지 머리를 쓰고 있는 세영이를 도와주자.

Unit 03 | 5주차 문제 소개

15기 정규세션
TOBIG'S 14기 김상현

Problem 2. 와인 컨퍼는 즐거워

첫째 줄에 와인 잔의 개수 n을 입력받는다. (1<=n<=10,000) 둘째 줄부터 n+1줄까지 와인 잔에 들어있는 와인의 양이 순서대로 주어진다. 와인의 양은 음이 아닌 정수이다.

최대로 마실 수 있는 와인의 양을 출력한다.

입력:
5
4
7
11
2
6

출력:
24

Unit 03 | 5주차 문제 소개

15기 정규세션
TOBIG'S 14기 김상현

Problem 3. 알록달록 울타리

저 푸른 초원 위에 그림 같은 집을 짓게 된 상현이는 알록달록한 울타리를 만들고 싶다.

울타리의 재료들은 높이가 모두 1m이고, 가로 길이는 색상에 따라 다르다. 울타리 재료들을 옆으로 붙여서 총 길이가 k m(미터)인 울타리를 만들 때, 울타리 재료의 경우의 수를 알고 싶다.

즉 n가지 색상의 재료가 있을 때, 재료들을 잘 이어 붙여서 길이의 합이 k m(미터)인 울타리를 만들 것이다. 이때, 각각의 재료들은 반복해서 사용할 수 있다. 또한, 재료의 구성이 같은데, 순서만 다른 것은 같은 경우이다.

Unit 03 | 5주차 문제 소개

15기 정규세션
TOBIG'S 14기 김상현

Problem 3. 알록달록 울타리

입력으로 첫째 줄에 n, k 가 주어진다. (1<=n<=100, 1<=k<=10,000) 다음 n 개의 줄에는 색상별 울타리 재료의 가로 길이가 주어진다. 가로 길이는 100,000보다 작거나 같은 자연수이다.

출력으로 첫째 줄에 경우의 수를 출력한다. 경우의 수는 2^{31}보다 작다.

입력:
3 10
1
2
5

출력:
10

경우 1

경우 2

Unit 03 | 5주차 문제 소개

코스 코드
593739b811

코스 링크
https://class.mimir.io/courses/593739b811/registrations/new

둘 중 아무 방식으로 참여 하시면 됩니다~!

들어주셔서 감사합니다.