문제 1. 상담 ( #14501 )

상담원 정우는 곧 퇴사를 하려고 합니다.

퇴사 날은 N+1일 이라서, 남은 N일 동안 상담을 해서 돈을 많이 벌어서 나가고 싶습니다.

가장 돈을 많이 받도록 상담을 골라서 신청했을 때, 그 금액을 출력하는 프로그램을 작성하세요.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1일 | 2일 | 3일 | 4일 | 5일 | 6일 | 7일 |
| T | 4 | 1 | 3 | 1 | 2 | 3 | 2 |
| P | 10 | 20 | 30 | 40 | 35 | 150 | 10 |

7

4 10

1 20

3 30

1 40

2 35

3 150

2 10

95

문제 2. 냅색 ( #12865 )

물건의 수 A와 배낭의 무게 B가 주어집니다.

순서대로 물건의 무게 W와 가치 V가 주어집니다.

배낭에 가장 가치있게 담았을 경우의 가치를 출력하는 프로그램을 작성하세요.

4 7

6 13

4 8

3 6

5 12

14

정답코드

(1)

import sys

sys.setrecursionlimit(99999999)

input = sys.stdin.readline

def sol(idx):

if idx > N+1:

return -9999999999999

if idx == N+1:

return 0

if dp[idx] != -1:

return dp[idx]

dp[idx] = max(sol(idx+1), sol(idx+table[idx][0]) + table[idx][1])

return dp[idx]

N = int(input())

table = [[] for \_ in range(N+1)]

for i in range(N):

a, b = map(int, input().split())

table[i+1] = [a, b]

dp = [-1 for \_ in range(N+1)]

ans = sol(1)

print(ans)

(2)

def recur(cur, w):

global ans

if w > m:

return -9999999

if cur == n:

return 0

if dp[cur][w] != -1:

return dp[cur][w]

dp[cur][w] = max(recur(cur+1, w + arr[cur][0]) + arr[cur][1], recur(cur + 1 , w))

return dp[cur][w]

n, m = map(int,input().split())

arr = [list(map(int,input().split())) for i in range(n)]

dp = [[-1 for \_ in range(100010)] for j in range(n)]

ans = recur(0,0)

print(ans)