



알고리즘 Week 5

20기 정규세션

TOBIG'S 19기 최다희

Contents



20기 정규세션
TOBIG'S 19기 최다희

Unit 01 | 3주차 과제 리뷰

Unit 02 | Dynamic Programming

Unit 03 | 5주차 과제 소개



20기 정규세션
TOBIG'S 19기 최다희

Unit 01 | 3주차 과제 리뷰

1. DNA

[입력 조건]

DNA의 수 N 과 문자열 길이 M 이 주어진다. N 은 1000보다 작거나 같은 자연수이고, M 은 50보다 작거나 같은 자연수이다.

[출력 조건]

첫째 줄에 Hamming Distance의 합(차이 나는 각 염기서열간 거리의 합)이 가장 작은 DNA를 출력하고, 둘째 줄에 Hamming Distance의 합을 출력하시오. 해당 DNA가 여러 개 있을 땐 사전순으로 가장 앞서는 것을 출력하시오.

[입력 예시]

5 8 #(DNA의 수, 문자열의 길이)
TATGATAC
TAAGCTAC
AAAGATCC
TGAGATAC
TAAGATGT

[출력 예시]

TAAGATAC
7



1. DNA

```
import sys
input = sys.stdin.readline

n,m = map(int,input().split())
arr = []
for _ in range(n):
    arr.append(input())

res = ''
cnt = 0

dna = ['A','C','G','T']

for i in range(m):
    tmp = [0,0,0,0]
    for j in range(n):
        tmp[dna.index(arr[j][i])] += 1
    res += dna[tmp.index(max(tmp))]
    cnt += (n-max(tmp))

print(res)
print(cnt)
```

```
import sys
input = sys.stdin.readline

N, M = map(int, input().split())
DNA = [input().rstrip() for _ in range(N)]

s = []
h_dis = 0
for i in range(M):
    dna = {'A': 0, 'C': 0, 'G': 0, 'T': 0}
    for j in range(N):
        dna[DNA[j][i]] += 1

    maxDNA = max(dna.values())
    for k in dna:
        if dna[k] == maxDNA and len(s) <= i:
            s.append(k)
        else:
            h_dis += dna[k]

print(''.join(s))
print(h_dis)
```



2. 오목

[입력 조건]

흑돌이 1, 백돌이 2, 두어지지 않은 칸이 0으로 주어진 19x19 바둑판이 주어진다.

[출력 조건]

첫 줄에 흑 승리 시 1, 백 승리 시 2, 아직 승부가 결정되지 않았을 시 0을 출력한다(육목 인정 X).
흑/백 승리 시 연속된 바둑돌 중 좌상단 끝에 있는 바둑돌의 좌표를 아래와 같이 출력한다.

[입력 예시]

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 2 0 0 2 2 2 1 0 0 0 0 0 0 0 0 0 0
0 0 1 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 2 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

[출력 예시]

1 #(흑 승리)
3 2 #(3행 2열)

2. 오목

```
from collections import deque
```

```
arr = []  
for _ in range(19):  
    line = list(map(int, input().split()))  
    arr.append(line)
```

```
visit = [  
    [[0] * 19 for _ in range(19)], # 우  
    [[0] * 19 for _ in range(19)], # 우하  
    [[0] * 19 for _ in range(19)], # 하  
    [[0] * 19 for _ in range(19)], # 우상  
]
```

```
def baduk(x,y,color,n,que,arr,d):
```

```
    dx = [1,1,0,1]  
    dy = [0,1,1,-1]
```

```
    ix = x + dx[d]  
    iy = y + dy[d]
```

```
    if (ix>=0) & (ix<19) & (iy>=0) & (iy<19) : # 이동가능한경우  
        if arr[iy][ix]==color:  
            if visit[d][iy][ix] == 0:  
                que.append([ix,iy,color])  
                visit[d][iy][ix] = n+1
```

```
    if len(que) > 0 :  
        nx, ny, color = que.popleft()  
        return baduk(nx,ny,color,n+1,que,arr,d)  
    else : return n
```

```
px, py, pw = 0, 0, 0
```

```
for x in range(0,19):  
    for y in range(0,19):  
        for d in range(4):  
            if arr[y][x] == 0 : continue  
            else :  
                que = deque([])  
                tmp = baduk(x,y,arr[y][x],2,que,arr,d)  
                if tmp == 6 : px, py, pw = x , y, arr[y][x]
```

```
if pw != 0 : print(pw, '\n', py+1, ' ', px+1, sep='')  
else : print(0)
```

Unit 01 | 3주차 과제 리뷰



20기 정규세션
TOBIG'S 19기 최다희

2. 오목

```
ans = 0
for i in range(n):
    for j in range(n):
        if graph[i][j] != 0:
            a = sol(i, j, graph[i][j])
            if a > 0:
                ans = (a, i+1, j+1)

if ans == 0:
    print(0)
else:
    print(ans[0])
    print(ans[1], ans[2])
```

```
import sys
input = sys.stdin.readline
dx = [1, 0, 1, -1]
dy = [0, 1, 1, 1]
graph = []
n = 19
for _ in range(n):
    graph.append(list(map(int, input().split())))

def sol(x, y, color):
    for i in range(4):
        cnt = 1
        nx = x
        ny = y
        while True:
            nx += dx[i]
            ny += dy[i]
            if 0 > nx or nx >= n or ny < 0 or ny >= n:
                break
            if graph[nx][ny] != color:
                break
            else:
                cnt += 1
        if cnt == 5:
            if graph[x-dx[i]][y-dy[i]] != color or x == 0:
                if color == 1:
                    return 1
                else:
                    return 2
    return 0
```


3. 치킨배달

[입력 조건]

첫째 줄에 정사각형 도시의 한 변의 길이 N 과 남겨두어야 할 치킨집의 수 M 이 주어진다.
둘째 줄부터 N 개의 줄에는 도시의 정보가 주어진다. 0은 빈 칸, 1은 집, 2는 치킨집이다.

[출력 조건]

도시에서 M 개의 치킨집만을 남겨야 한다. 어떻게 남겨야 각 집과 치킨집의 격자 거리(맨하탄 거리)를 최소화할 수 있는가? 최솟값을 출력하시오.

[입력 예시]

```
5 3
0 0 1 0 0
0 0 2 0 1
0 1 2 0 0
0 0 1 0 0
0 0 0 0 2
```

[출력 예시]

5



3. 치킨배달

```
import sys
from itertools import combinations

input = sys.stdin.readline

n, m = map(int, input().split())
city = list(list(map(int, input().split())) for _ in range(n))
res = 999999
house = []
chicken = []

for i in range(n):
    for j in range(n):
        if city[i][j] == 1:
            house.append([i, j])
        elif city[i][j] == 2:
            chicken.append([i, j])

for chi in combinations(chicken, m):
    temp = 0
    for h in house:
        chi_len = 999
        for j in range(m):
            chi_len = min(chi_len, abs(h[0] - chi[j][0]) + abs(h[1] - chi[j][1]))
        temp += chi_len
    res = min(res, temp)

print(res)
```



20기 정규세션
TOBIG'S 19기 최다희

Unit 02 | Dynamic Programming

Dynamic Programming (동적계획법) 이란?

큰 문제를 여러 개의 작은 문제로 나누어 푸는 방법

Dynamic Programming 방법

모든 작은 문제를 한번만 풀기 위해서, Memoization을 통해 처리

Memoization ?

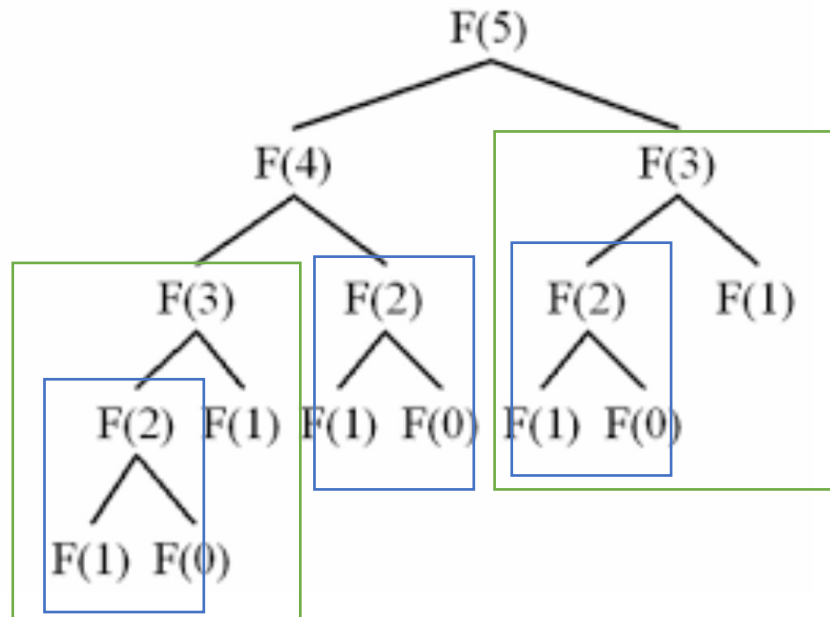
작은 문제부터 해결하고 그 결과를 저장해서 다시 큰 문제를 해결할 때 사용

Unit 02 | Dynamic Programming



20기 정규세션
TOBIG'S 19기 최다희

1. 작은 문제들이 반복된다.
2. 같은 문제는 구할 때 마다 정답이 같다.



$$F(5) = F(4) + F(3)$$

$$F(5) = F(3) + F(2) + F(3)$$

Unit 02 | Dynamic Programming



20기 정규세션
TOBIG'S 19기 최다희

1. Top-down

```
#include <stdio.h>

int dp[100] = {0,};    // 하위 문제 답을 저장할 메모이제이션 배열

int fibo(int n) {
    if (n <= 1) {
        return n;
    } else {
        if (dp[n] > 0) {           // 해당 문제의 답이 존재
            return dp[n];
        }
        dp[n] = fibo(n-1) + fibo(n-2);
        return dp[n];
    }
}

int main(){
    printf("%d", fibo(5));
    return 0;
}
```

재귀적인 함수 호출



Unit 02 | Dynamic Programming



20기 정규세션
TOBIG'S 19기 최다희

2. Bottom-up

```
int dp[100] = {0,};    // 하위 문제 답을 저장할 메모이제이션 배열

int fibo(int n) {
    dp[0] = 0;
    dp[1] = 1;
    int i;
    for (i=2; i<=n; i++) {    // 2부터 시작해서 n까지 반복
        dp[i] = dp[i-1] + dp[i-2];
    }
    return dp[n];
}

int main(){
    printf("%d", fibo(5));
    return 0;
}
```

반복문 활용



Colored by Color Scriptor CS



20기 정규세션
TOBIG'S 19기 최다희

Unit 03 | 5주차 과제 소개

Unit 03 | 5주차 과제 소개



20기 정규세션
TOBIG'S 19기 최다희

1. 피보나치 수 2 (2748번)

<https://www.acmicpc.net/problem/2748>

문제

피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.

이를 식으로 써보면 $F_n = F_{n-1} + F_{n-2}$ ($n \geq 2$)가 된다.

$n=17$ 일때 까지 피보나치 수를 써보면 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597

n 이 주어졌을 때, n 번째 피보나치 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 n 이 주어진다. n 은 90보다 작거나 같은 자연수이다.

출력

첫째 줄에 n 번째 피보나치 수를 출력한다.

예제 입력 1 복사

10

예제 출력 1 복사

55

Unit 03 | 5주차 과제 소개

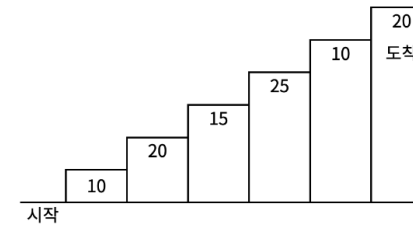


20기 정규세션
TOBIG'S 19기 최다희

2. 계단오르기 (2579번)

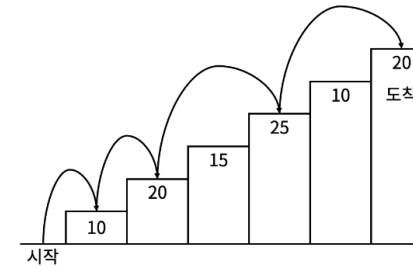
<https://www.acmicpc.net/problem/2579>

계단 오르기 게임은 계단 아래 시작점부터 계단 꼭대기에 위치한 도착점까지 가는 게임이다. <그림 1>과 같이 각각의 계단에는 일정한 점수가 쓰여 있는데 계단을 밟으면 그 계단에 쓰여 있는 점수를 얻게 된다.



<그림 1>

예를 들어 <그림 2>와 같이 시작점에서부터 첫 번째, 두 번째, 네 번째, 여섯 번째 계단을 밟아 도착점에 도달하면 총 점수는 $10 + 20 + 25 + 20 = 75$ 점이 된다.



<그림 2>

계단 오르는 데는 다음과 같은 규칙이 있다.

1. 계단은 한 번에 한 계단씩 또는 두 계단씩 오를 수 있다. 즉, 한 계단을 밟으면서 이어서 다음 계단이나, 다음 다음 계단으로 오를 수 있다.
2. 연속된 세 개의 계단을 모두 밟아서는 안 된다. 단, 시작점은 계단에 포함되지 않는다.
3. 마지막 도착 계단은 반드시 밟아야 한다.

따라서 첫 번째 계단을 밟고 이어 두 번째 계단이나, 세 번째 계단으로 오를 수 있다. 하지만, 첫 번째 계단을 밟고 이어 네 번째 계단으로 올라가거나, 첫 번째, 두 번째, 세 번째 계단을 연속해서 모두 밟을 수는 없다.

각 계단에 쓰여 있는 점수가 주어질 때 이 게임에서 얻을 수 있는 총 점수의 최댓값을 구하는 프로그램을 작성하시오.

Unit 03 | 5주차 과제 소개



20기 정규세션
TOBIG'S 19기 최다희

입력

입력의 첫째 줄에 계단의 개수가 주어진다.

둘째 줄부터 한 줄에 하나씩 제일 아래에 놓인 계단부터 순서대로 각 계단에 쓰여 있는 점수가 주어진다. 계단의 개수는 300이하의 자연수이고, 계단에 쓰여 있는 점수는 10,000이하의 자연수이다.

출력

첫째 줄에 계단 오르기 게임에서 얻을 수 있는 총 점수의 최댓값을 출력한다.

예제 입력 1 [복사](#)

6
10
20
15
25
10
20

예제 출력 1 [복사](#)

75

Unit 03 | 5주차 과제 소개



20기 정규세션
TOBIG'S 19기 최다희

3. 정수삼각형 (1932번)

<https://www.acmicpc.net/problem/1932>

문제

```
      7
     3 8
    8 1 0
   2 7 4 4
  4 5 2 6 5
```

위 그림은 크기가 5인 정수 삼각형의 한 모습이다.

맨 위층 7부터 시작해서 아래에 있는 수 중 하나를 선택하여 아래층으로 내려올 때, 이제까지 선택된 수의 합이 최대가 되는 경로를 구하는 프로그램을 작성하라. 아래층에 있는 수는 현재 층에서 선택된 수의 대각선 왼쪽 또는 대각선 오른쪽에 있는 것 중에서만 선택할 수 있다.

삼각형의 크기는 1 이상 500 이하이다. 삼각형을 이루고 있는 각 수는 모두 정수이며, 범위는 0 이상 9999 이하이다.

입력

첫째 줄에 삼각형의 크기 n ($1 \leq n \leq 500$)이 주어지고, 둘째 줄부터 $n+1$ 번째 줄까지 정수 삼각형이 주어진다.

출력

첫째 줄에 합이 최대가 되는 경로에 있는 수의 합을 출력한다.

예제 입력 1 복사

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

예제 출력 1 복사

30

Unit 03 | 5주차 과제 소개



20기 정규세션
TOBIG'S 19기 최다희

4. 퇴사(14501번)

<https://www.acmicpc.net/problem/14501>

상담원으로 일하고 있는 백준이는 퇴사를 하려고 한다.

오늘부터 $N+1$ 일째 되는 날 퇴사를 하기 위해서, 남은 N 일 동안 최대한 많은 상담을 하려고 한다.

백준이는 비서에게 최대한 많은 상담을 잡으라고 부탁을 했고, 비서는 하루에 하나씩 서로 다른 사람의 상담을 잡아놓았다.

각각의 상담은 상담을 완료하는데 걸리는 시간 T_i 와 상담을 했을 때 받을 수 있는 금액 P_i 로 이루어져 있다.

$N = 7$ 인 경우에 다음과 같은 상담 일정표를 보자.

	1일	2일	3일	4일	5일	6일	7일
T_i	3	5	1	1	2	4	2
P_i	10	20	10	20	15	40	200

1일에 잡혀있는 상담은 총 3일이 걸리며, 상담했을 때 받을 수 있는 금액은 10이다. 5일에 잡혀있는 상담은 총 2일이 걸리며, 받을 수 있는 금액은 15이다.

상담을 하는데 필요한 기간은 1일보다 클 수 있기 때문에, 모든 상담을 할 수는 없다. 예를 들어서 1일에 상담을 하게 되면, 2일, 3일에 있는 상담은 할 수 없게 된다. 2일에 있는 상담을 하게 되면, 3, 4, 5, 6일에 잡혀있는 상담은 할 수 없다.

또한, $N+1$ 일째에는 회사에 없기 때문에, 6, 7일에 있는 상담을 할 수 없다.

퇴사 전에 할 수 있는 상담의 최대 이익은 1일, 4일, 5일에 있는 상담을 하는 것이며, 이때의 이익은 $10+20+15=45$ 이다.

상담을 적절히 했을 때, 백준이가 얻을 수 있는 최대 수익을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N ($1 \leq N \leq 15$)이 주어진다.

둘째 줄부터 N 개의 줄에 T_i 와 P_i 가 공백으로 구분되어서 주어지며, 1일부터 N 일까지 순서대로 주어진다. ($1 \leq T_i \leq 5$, $1 \leq P_i \leq 1,000$)

출력

첫째 줄에 백준이가 얻을 수 있는 최대 이익을 출력한다.



5. 동전1 (2293번)

<https://www.acmicpc.net/problem/2293>

문제

n 가지 종류의 동전이 있다. 각각의 동전이 나타내는 가치는 다르다. 이 동전을 적당히 사용해서, 그 가치의 합이 k 원이 되도록 하고 싶다. 그 경우의 수를 구하시오. 각각의 동전은 몇 개라도 사용할 수 있다.

사용한 동전의 구성이 같은데, 순서만 다른 것은 같은 경우이다.

입력

첫째 줄에 n, k 가 주어진다. ($1 \leq n \leq 100, 1 \leq k \leq 10,000$) 다음 n 개의 줄에는 각각의 동전의 가치가 주어진다. 동전의 가치는 100,000보다 작거나 같은 자연수이다.

출력

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

예제 입력 1 [복사](#)

```
3 10
1
2
5
```

예제 출력 1 [복사](#)

```
10
```



5주차 알고리즘 과제 기한 : 8월 30일 20:00 (2주)

질문/힌트/모든 문의는 19기 김은지/ 위성진/ 이동준/ 임서영 최다희/ 표지원/ 한진모 에게

언제든 자유롭게 해주세요!



20기 정규세션
TOBIG'S 19기 최다희

