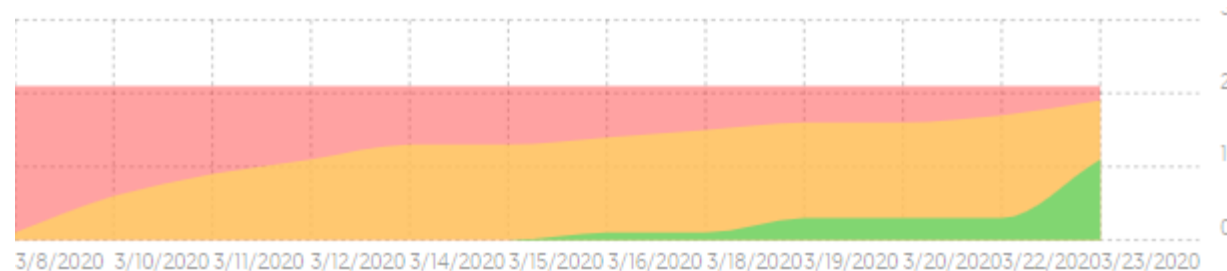


13기 정규세션

ToBig's 12기 윤승아

Algorithm Week9

ce9164357e



Contents

Unit 01 | Week7 풀이

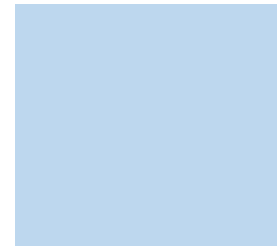
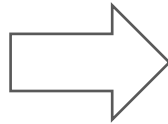
Unit 02 | Coursecode : [ce9164357e](#)

Unit 03 | Q & A

Unit 01 | Week7 풀이

Problem 1. 새로운 길만을 개척하는 이승현

현재 노드에서 다음 노드로 갈 때



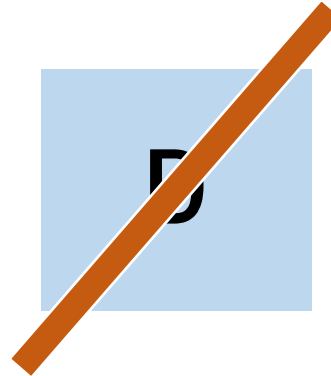
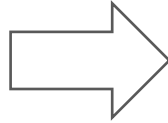
{현재노드까지 지나오면서 모은 알파벳}

{A, C, D}

Unit 01 | Week7 풀이

Problem 1. 새로운 길만을 개척하는 이승현

If 현재 노드에서 다음 노드로 갈 때 이미 포함하는 알파벳일 때

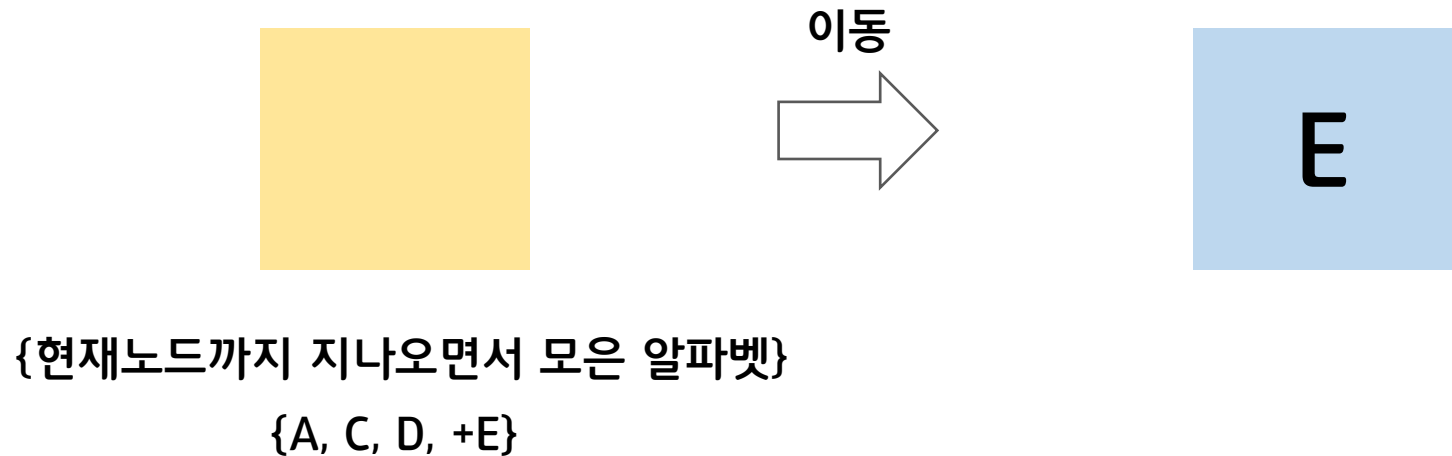


{현재노드까지 지나오면서 모은 알파벳}
{A, C, D}

Unit 01 | Week7 풀이

Problem 1. 새로운 길만을 개척하는 이승현

If 현재 노드에서 다음 노드로 갈 때 이미 포함하지 않는 알파벳일 때



Unit 01 | Week7 풀이

Problem 1. 새로운 길만을 개척하는 이승현

코드

```
dx = [-1, 1, 0, 0]
dy = [0, 0, 1, -1]
routine = [0] * 26
```

```
def forward(row, column, routine): # 현 위치의 행과 열 인덱스와 지난 알파벳 리스트
    ret = 0
    for i in range(4):
        next_row = row + dx[i]
        next_col = column + dy[i]
        if (0 <= next_col < num_column) and (0 <= next_row < num_row):
            alphabet = board[next_row][next_col]
            if routine[ord(alphabet)-65] == False:
                routine[ord(alphabet)-65] = True
                ret = max(ret, forward(next_row, next_col, routine)+1)
            routine[ord(alphabet)-65] = False
    return ret
routine[ord(board[0][0])-65] = True
answer = forward(0, 0, routine) + 1
```

Unit 01 | Week7 풀이

Problem 2. 회장뽑

음 완전탐색.

```
import itertools
N = int(input())
arr = []
for i in range(N):
    arr.append(list(map(int, input().split())))
K = list(range(N))
ret = 50000
ans = []
for k in range(1, len(K)):
    for comb in itertools.combinations(K, k):
        con=0
        for cx in comb:
            for cy in comb:
                con+=arr[cx][cy]
        ans.append(con)
for i in range(0, len(ans)):
    ret = min(ret, abs(ans[i]-ans[-i-1]))
print(ret)
```

Unit 01 | Week7 풀이

Problem 3. 혼자 앉고 싶은 이승현

점화식을 세우는 문제, $Dp(n)$ 은 $2*n$ 사각형에 앉는 경우의 수

$$Dp(n) = Dp(n-1) + ???$$

점화식을 아까전처럼 세우면 망한다. 경우의 수는 아래와 같이 3가지가 있다.

$$Dp(n-1, 2)$$
$$Dp(n-1, 1)$$
$$Dp(n-1, 0)$$

Unit 01 | Week7 풀이

Problem 3. 혼자 앉고 싶은 이승현

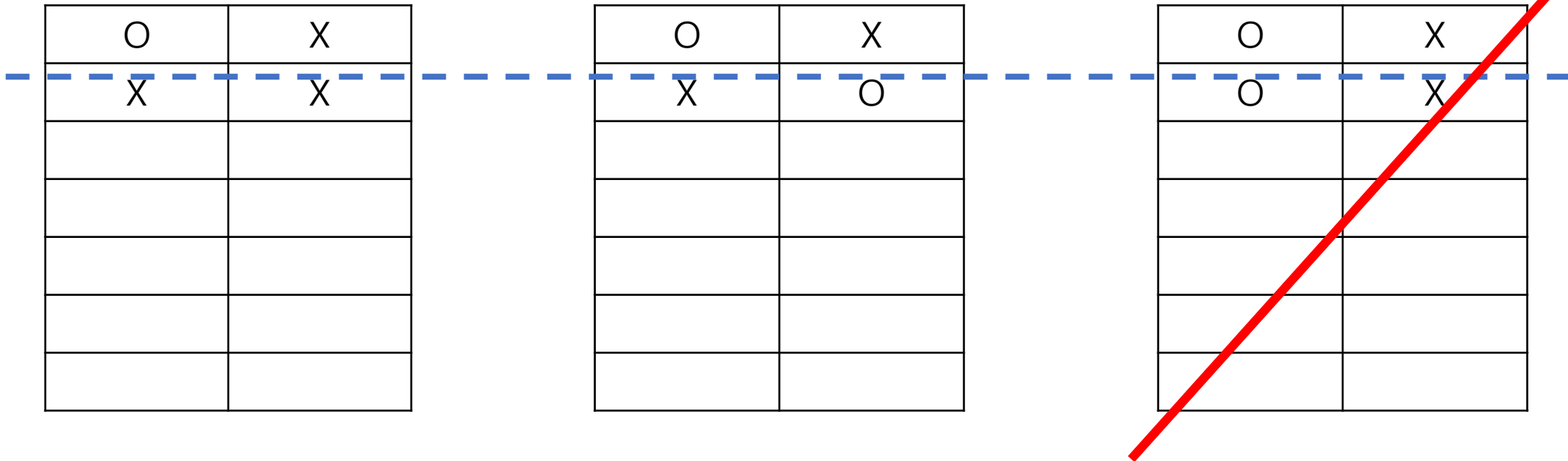
DP(n, 0) 때 $Dp(n, 0) = Dp(n-1, 0) + Dp(n-1, 1) + Dp(n-1, 2)$ 왜?

X	X		X	X		X	X
X	X		X	O		O	X

Unit 01 | Week7 풀이

Problem 3. 혼자 앉고 싶은 이승현

DP(n, 1) 때 $Dp(n, 1) = Dp(n-1, 0) + Dp(n-1, 2)$ 왜?



DP(n, 2) 때 $DP(n, 2) = DP(n-1, 0) + DP(n-1, 1)$ 왜?

X	O
X	X

X	O
O	X

X	O
X	O

Unit 01 | Week7 풀이

Problem 3. 혼자 앉고 싶은 이승현

갑분 C코드

```
dp[1][0] = 1, dp[1][1] = 1, dp[1][2] = 1;

for (i = 2; i <= N; i++) {
    dp[i][0] = (dp[i-1][0] + dp[i-1][1] + dp[i-1][2]) % MOD;
    dp[i][1] = (dp[i-1][0] + dp[i-1][2]) % MOD;
    dp[i][2] = (dp[i-1][0] + dp[i-1][1]) % MOD;
}
```

Unit 01 | Week7 풀이

Problem 3. 혼자 앉고 싶은 이승현

또다른 풀이 : 점화식 때려박기

1 2 3 4...

$$dp(n) = dp(n-1) * 2 + dp(n-2)$$

주의사항 : 패턴이 보인다고 해인 것은 절대! 아니다.
규칙을 찾지 말고 해를 찾자!

3 7 17 41...

Unit 01 | Week7 풀이

Problem 4. 수학을 좋아하는 이승현

간단한 풀이 : 수학 잘하는 것처럼 풀기

$$GCF = (100)*G + (10) * C + F$$

$$ACDEB = (10000)*A + (1000) * C + (100)*D + (10)*E + B$$

$$+ = (10000)*A + (1010)*C + (100)*D + (100)*G + (10)*E + B + F$$

큰숫자대로 넣자!

Unit 01 | Week7 풀이

Problem 4. 수학을 좋아하는 이승현

복잡한 풀이 : 완전탐색

Unit 01 | Week7 풀이

이번주에 풀어보신 알고리즘 문제 난이도

1번



Gold III

퇴각 검색 비트마스킹

2번



Gold V

완전 탐색

3번



Silver I

동적 계획법

4번



Gold IV

퇴각 검색 탐욕법

고생하셨습니다.

Q & A

들어주셔서 감사합니다.