[10월 1주차]

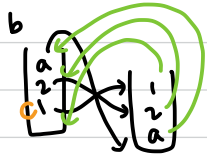
9935) 폭발 → stack

그냥 하면 $O(10^6/36 * N)$ → TLE

Stack 말고 erase로 더 간단하게 풀수도있음!

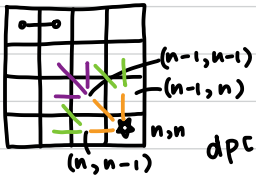
ret += s[i] 를해가면서 ret.size ≥ 폭발문자열size 일때 substr(ret.size - △, ret.size)랑 폭발이랑 비교 → 같으면 erase!

폭발의 { 마지막 문자와 제외하고 다 push
{ 마지막 문자면 폭발문자열 size-1 만큼
top 비교 ⇒ 폭발문자열[i]에해당⇒ pop, stk2에저장
(해당 x → stk2 다시다 옮기기



풀이) i=1, j=1부터 벗어나가는 방식으로 해도됨!

11060) $3 \le N \le 16$



현상태:
0 1 2
— | \

dp[n][n][0] + dp[n][n][1] + dp[n][n][2]
=
dp[n][n-2][0] + dp[n-1][n-2][2]
dp[n-2][n-1][2] + dp[n-2][n][1]
dp[n-2][n-2][2] + dp[n-1][n-2][0] + dp[n-2][n-1][1]

(초기)
dp[1][2][0]=1
dp[1][3][0]=1
dp[2][3][2]=1
dp[3][3][1]=1

Tip)
DP를 항상
dp[1][1][0]
[1]
[2]

for(int i=3; ~)
for(int j=3; ~)

dp[i][j] = dp[i-✱][j-△] + …
현재 이전

이렇게 뺄셈으로 이전값을 생각하는 방식이 아니라

for(int i=1; ~)
for(int j=1; ~)

dp[i+♡][j+□] += dp[i][j]
다음 현재

다음으로 벗어나가는 방식 (+= 현재) 도 가능! (이게 더쉬운듯)

4179) 저번에 풀다 만 앗네 ㅎ

시간복잡도 계산?

1) fire 배열 따로 만들어서
옆불에 도착하는지 ▦ 적어두고

2) 지훈이 DFS

✱ 풀이) 지훈이의 각 칸에도착하는 최단거리 J
(불의 " 최단거리 비교 F
↝ J<F 면 이동가능 (지훈이가)

✱ 풀이) 불 확산하는방법

① 처음에 입력받을때 'F'(불)인칸이면, 큐(queue)에 삽입!

② 이후 큐에 원소있을동안 = while(q.size())
4방향으로 dy,dx → 만약 or(dC[ny][nx]!=INF continue 벽일경우 이미이전에 최단거리가 계산된칸인경우
갈수있는칸이면 dC[ny][nx]=dC[y][x]+1 하고 ㉤에 push

⇓

이렇게하면 1분때 확산, 2분때 확산, … n분때 확산
순서대로 처리 가능! (큐 니까)

1781) day 1 2 3 4 . .

④ ① ③ ① ⑥ ② ⑤

데드라인 ≤N 물품# 1 2 3 4 5 6 7
데드라인 1 ~~1~~ ~~3~~ 3 2 ~~2~~ 6
컵라면수 6 7 2 1 4 5 1

1) "구간"에 대한거니까 정렬 후 뭐 해야겠다 ~ 생각 (데드라인 기준. 그래야 배치생각안해도 ok)
2) 힙 을 만드는방법 < 최대를 더 크게 / 최소를 더 작게 (V)
3) 우선순위큐 (힙) 사용

현상태 y, x 에도달하기까지 움직인횟수

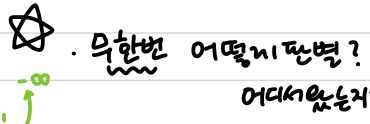
go(int y, int x 동전움직인

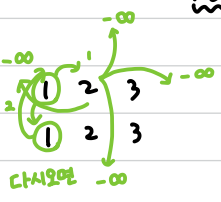
1103) dp[ ] : 횟수

기저사례 : 구멍 return -1e9;
보드바깥 (처음)

memoization int &ret =
if (ret != 초기값) return ret;

리그





네방향순회전후에 check 배열 < true false

16637) (3+(2)*4)

< 모든 case 만들기
계산 → 수식순회 → ')' 나오면
앞3개계산, 앞2개뒤부터
똑같이 반복 (idx--) 빼고

0 2 4 6
(3+(8)*(7)-(9)*2) 하나 건너뛰어야함

go(int idx, check

기저사례 if (idx >= n) → check[idx]=1 인거
[계산→최대갱신
return

체크 check[idx]=1

로직 현재괄호 사용 or 현재 사용X
go(idx+4) go(idx+2)

원복 check[idx]=0