

SimJoon_TTMD

2021_04_21

동아리 2주차

SimJoon_TTMD

10835

- 한번의 쿼리에서 나올수 있는 경우를 다 고려하면서 계산하면 된다.
- i 는 왼쪽카드 개수, j 는 오른쪽 카드 개수
- 왼쪽 > 오른쪽 => $dp[i][j] = \max(\text{오른쪽값} + \text{오른쪽을 더하기 이전상태...})$ 으로 잘 표현해보자!

SimJoon_TTMD

2306

- 완전탐색 느낌이 나지만 그렇게 풀면 TLE가 뜬다!
- Dp로 시간을 줄여보자!
- $Dp[i][j]$ = $i \sim j$ 까지의 문자열에서의 최대값.
- $Dp[i][j] = dp[i][k] + dp[k+1][j]$
- $A \sim g$ 사이 == $T \sim c$ 사이, $dp[i][j] = dp[i+1][j-1] + 2$

SimJoon_TTMD

2515

- 이분탐색과 DP를 잘사용해보자
- $D[i]$ 를 $pic[0 \sim i]$ 를 전시할 때 최댓값(단 i 번째 그림은 반드시 보여짐)이라고 잡는다면, $D[i]$ 는 $pic[i]$ -S 이하의 높이를 가진 그림 $pic[0], pic[1], \dots, pic[k]$ 에 대해 $D[i] = \max(D[0], D[1], \dots, D[k]) + pic[i]$ 의 높이이다.
- $\max(D[0], D[1], \dots, D[k])$ 를 각 i 에 대해 매번 구하게 될 경우 $O(N^2)$ 이지만, \max 값을 누적시켜서 계속 가져가면 $O(N)$ 으로 해결이 가능합니다. 그러므로 시간복잡도는 최초의 정렬때 $O(N \lg N)$, 이후 DP에서 $O(N)$ 이 되어 $O(N \lg N)$ 이 된다.

SimJoon_TTMD

11058

- 손으로 써보자
1~6까지는 i 가 최대라는걸 알 수 있다.
- $\text{Ctrl}(a,c,v)$ 가 2배를 증가해주는 키(3개를 사용)
- $\text{Dp}[i] = \max(\text{dp}[i-3]*2, \text{dp}[i-4]*3, \text{dp}[i-5]*4)$ 라는걸 알수있다.

SimJoon_TTMD

10942

- $Dp[i][j]$ = $i \sim j$ 에서가 팰린드롬인지
- 갱신해주는 점화식을 생각해보자.
- $Arr[i] == arr[j] \ \&\& \ dp[i+1][j-1] == \text{팰린드롬}$ 이면 $dp[i][j]$ 는 뭘까?

SimJoon_TTMD

12865

- 유명한 01ナップ색 문제
- <https://huiyu.tistory.com/entry/DP-01-Knapsack%EB%B0%B0%EB%82%AD-%EB%AC%B8%EC%A0%9C>
- 한번 읽어보자!

SimJoon_TTMD

2618

- $Dp[x][y]$ = 1번경찰차가 x 번째 사건, 2번경찰차가 y번째 사건 해결했을때의 비용
- $Dp[x][y] = \min(Dp[\text{현재사건}][y] + \text{현재사건이랑 이전사건거리}, Dp[x][\text{현재사건}] + \text{현재사건이랑 이전사건거리})$