

Problem1

비교하는 문자열 2개를 각각 $X(x[0], x[1], x[2], \dots, x[i], \dots, x[a])$, $Y(y[0], y[1], y[2], \dots, y[j], \dots, y[b])$ 라고 하고 $S[i][j]$ 를 $x[i], y[j]$ 까지의 최대 유사도 점수라고 할 때 $S[i][j]$ 는 $S[i-1][j-1] + (s \text{ or } -f)$, $S[i-1][j]-p$, $S[i][j-1]-p$ 중 최댓값을 따라갑니다.

자세한 table setting은 아래식을 따라갑니다.

$$S(i, j) = \begin{cases} -2j & (i=0) \\ -2i & (j=0) \\ \max(S(i-1, j-1) + s(i, j), S(i-1, j) - 2, S(i, j-1) - 2) & (i > 0, j > 0) \end{cases}$$

Problem2

arr을 문자열이라 하고 $S[i][j]$ 를 $i \sim j$ 까지가 회문인지 아닌지를 저장하는 table이라고 할 때 $i=j$ 이거나 $arr[i]$ 와 $arr[j]$ 가 같을 때 한칸 차이나거나($j-i=1$) 그 사이가 회문이면 $S[i][j]=1$ 로 저장합니다. 자세한 식은 아래와 같습니다.

$$S(i, j) = \begin{cases} 1 & \text{if } i=j \text{ or } (arr[i]=arr[j] \text{ and } (S(i-1, j-1)=1 \text{ or } j-i=1)) \\ 0 & \text{else} \end{cases}$$

Problem3

두 카드 쌍을 각각 $X(x[0], x[1], x[2], \dots, x[i], \dots, x[a])$, $Y(y[0], y[1], y[2], \dots, y[j], \dots, y[b])$ 라고 하고 $S[i][j]$ 를 x 에서 i 장 y 에서 j 장 꺼냈을 때 만들 수 있는지 없는지를 0, 1로 저장하는 table이라고 할 때 $S[i-1][j-1]=1$ 일때 $x \sim y$ 에서 testcase에 맞는 카드쌍이 있다면 $S[i][j-1]$ 혹은 $S[i-1][j] = 1$ 로 나아가게 됩니다. 운 좋게 양쪽 다 알맞은 카드가 있다면 $S[i][j-1]$, $S[i-1][j]$ 모두 1이 됩니다. 그렇게 칸을 채워나가면 testcase에 부합하는 카드쌍이면 $S[1][1] \sim S[i][j]$ 까지 1로 마킹된 길이 생깁니다. 그 경우엔 true(1)을 return 해줍니다. 식을 아래와 같습니다.

$$S(i, j) = \begin{cases} 1 & \text{if } \left(\begin{array}{l} S(i-1, j) = 1 \text{ and } \text{testcase}[i+j-1] = x[i-1] \\ \text{or} \\ S(i, j-1) = 1 \text{ and } \text{testcase}[i+j-1] = y[j-1] \end{array} \right) \\ 0 & \text{else} \end{cases}$$