

Estructuras de Datos y Algoritmos II

Tarea 07

Rubén Pérez Palacios
Profesor: Dr. Carlos Segura González

15 Marzo 2020

Mi solución al problema es una DP, la cual tiene tres estados:

- Renglón i
- Columna j
- Máscara de bits $mask$, donde el j –ésimo bit representa si esta o no ocupado la casilla $c_{i+1,j}$ si $j < c$ o la casilla $c_{i,j}$ si $j \geq c$.

Donde $dp(i, j, mask)$ respresenta el total de acomodos que son posibles hacer en tablero de $N \times M$, donde estan ocupadas las casillas $c_{i,j}$ tales que $0 \leq i < N, 0 \leq j < M$, además de las ya mencionadas por la bitmask (si $j < c$, todas las casillas $c_{i,j}$ también se consideran ocupadas).

Para no hacer uso de una memoria excesiva debemos tener cuidado. Por como está definida nuestra mascara debe contener M bits pero esto puede ser muy grande, pero tranquilos ya que nos damos cuenta que ya sea N o M alguno siempre será menor o igual a 10, por lo que podemos intercambiar a N y M si $N < M$ de este modo aseguraremos que $M \leq 10$, y por lo tanto la complejidad de nuestra memoria es

$$O(NM2^{\min(N,M)}).$$

También debemos tener cuidado en hacer operaciones inecesaria y que están hagan que nuestra complejidad de tiempo se dispare. Debido a que el problema es multitask, no podemos inicializar la matriz por cada caso, ya que nuestra complejidad sería de $O(TNM2^{\min(N,M)})$, por lo que haremos uso de un matriz auxiliar para evitarlo, por lo que nuestra complejidad final sería

$$O(NM2^{\min(N,M)}).$$