

1.º recuperatorio de examen parcial – 26/02/2024

1. Se tiene una matriz de $n \times m$ casilleros, en la cual empezamos en la posición $(0, 0)$ (arriba a la izquierda) y queremos llegar a la posición $(n - 1, m - 1)$ (abajo a la derecha), pero solamente nos podemos mover hacia abajo o hacia la derecha, y comenzamos con una vida inicial V . Cada casillero puede estar vacío, o tener una trampa. En los casilleros que hay trampas se nos reduce la vida en una cantidad T_i conocida (dependiente de cada casillero).

Diseñar un algoritmo de programación dinámica que dados todos los datos necesarios, permita determinar la cantidad de vida máxima con la que podemos llegar a $(n - 1, m - 1)$. Implementar también una forma de poder reconstruir dicho camino. Indicar la complejidad del algoritmo propuesto, en tiempo y espacio.

2. Todos los años la asociación de Tiro con Arco profesional realiza una preclasificación de los n jugadores que terminaron en las mejores posiciones del ranking para un evento exclusivo. En la tarjeta de invitación quieren adjuntar el número de posición en la que está actualmente y a cuántos rivales invitados logró superar en el ranking, en comparación al ranking del año pasado. Contamos con un listado que tiene el nombre del jugador y la posición del ranking del año pasado ordenado por el ranking actual. Implementar un algoritmo que dada la lista mencionada, devuelva (por ejemplo, en un diccionario) a cuántos rivales ha superado cada uno de los invitados. Para realizar esto de forma eficiente, recomendamos utilizar División y Conquista.

Ejemplo: LISTA: [(A, 3), (B, 4), (C, 2), (D, 8), (E, 6), (F, 5)]. Se puede ver que el jugador A pasó del 3er lugar al 1er lugar, superando al jugador C. El jugador B llegó al segundo lugar y superó al jugador C. El jugador C no logró superar a ninguno de los invitados (si bien se encuentra en la tercera posición, ya tenía el año anterior mejor clasificación que el resto de invitados, por tanto no logró superar a ninguno), etc...

3. El K-core de un grafo es el subgrafo del mismo en el cuál todos los vértices tienen grados mayor o igual a K. Implementar un algoritmo greedy que dado un grafo y un valor K devuelva el K-core del grafo (es decir, el subgrafo en el cuál todos los vértices involucrados tienen grado mayor o igual a K, en dicho subgrafo). Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy.
4. Implementar un algoritmo que resuelva el problema de 3-SAT: determinar si, dado un grupo de cláusulas de 3 términos (pudiendo ser complementos de variables), existe alguna asignación de valores de las variables tal que la disyunción de todas las cláusulas (que son todas conjunciones) evalúan a **true**.
5. Un director de teatro tiene un elenco de n actores para realizar sus obras (todos los actores se encuentran capacitados para ello). Para cada obra necesita diferente cantidad de actores. Para evitar problemas en el elenco, decide que nunca repetirá para una obra 2 actores que ya hayan participado juntos en una obra previa. El problema de los actores dice: “Dado un conjunto de n actores y la información de en cuáles obras actuaron, ¿existe forma de seleccionar k de ellos para hacer la siguiente obra, sin que existan dos de ellos que hayan compartido elenco previamente?”.

Demostrar que el problema de los actores es un problema NP-Completo.