

Final de Programación 3 - TUDAI - 18/03/2024

Ejercicio 1

Considere el fragmento de código de la derecha donde A es un array de números enteros. Responda:

- Determinar qué función cumple el código.
- ¿Cuál es su complejidad temporal?
- ¿Es posible realizarlo de forma más eficiente? Justifique.

```
int i, j, aux;
for(i=0; i<A.length-1; i++)
    for(j=0; j<A.length-i-1; j++)
        if(A[j+1]<A[j]){
            aux=A[j+1];
            A[j+1]=A[j];
            A[j]=aux;
        }
```

Ejercicio 2

Escriba un algoritmo en JAVA que dado un árbol binario y un valor X pasado como parámetro, devuelva, si existen, todas las ramas cuya suma sea mayor o igual a X.
Una rama es el conjunto de nodos en un camino desde la raíz hacia alguna hoja.

Ejercicio 3

Dada una matriz de números enteros positivos de tamaño NxN, debemos encontrar un camino que incluya la mayor cantidad de celdas de la matriz teniendo en cuenta que:

- Se debe seleccionar desde qué celda iniciar el recorrido (la celda inicio NO viene por parámetro)
- Solo podemos ir de una celda a otra celda adyacente en dirección horizontal y vertical.
- Solo podemos ir de una celda a otra celda que contenga un valor menor o igual al valor de la celda actual.
- No podemos pasar más de una vez por la misma celda.

Plantee un pseudo-Java que resuelva el problema utilizando la técnica greedy, y describa con sus palabras la estrategia utilizada.

Ejercicio 4

Dado un número entero positivo N y un valor objetivo V se desean encontrar todas las posibles combinaciones de números entre 1 y N cuya suma de como resultado V. Los números utilizados pueden aparecer más de una vez dentro de una combinación, pero no pueden aparecer más de 4 veces.

No deben aparecer combinaciones duplicadas en el resultado final.

Por ejemplo, si N= 3 y V= 5, la solución sería { [1,1,1,2], [1, 1, 3], [1, 2, 2], [2, 3] }

Casos no válidos: por ej. las combinaciones [1,1,1,2] y [1,2,1,1] son consideradas duplicadas, la solución [1,1,1,1,1] no sería posible ya que el 1 se repite más de 4 veces y eso no está permitido.

- Dibuje el árbol de búsqueda con todas las marcaciones que considere necesarias.
- Plantee un pseudo-código Java que resuelva el problema mediante la técnica de Backtracking.

Ejercicio 5

- Responda:

i- Si nos piden diseñar un algoritmo para resolver un problema real que es parte de la operación de un sistema (ej. reservas de vuelos), ¿La opción de hacerlo mediante backtracking siempre es una alternativa válida de diseño de algoritmo para la solución? Justifique su respuesta.

ii- ¿Cuál será la altura mínima de un árbol binario de búsqueda de 5000 elementos? deje expresado si hace algún cálculo.

iii- ¿Cuál será la altura máxima de un árbol binario de búsqueda de 5000 elementos? deje expresado si hace algún cálculo.

- A cada afirmación responda si es Verdadera o Falsa. Justifique las que considere Falsas.

i- Si se hace un DFS sobre un grafo G, siempre que se encuentre un arco que me lleve a un vértice por el cual ya pasó el algoritmo habremos detectado un ciclo.

ii- La matriz de adyacencia de un grafo no dirigido siempre será simétrica.

iii- El algoritmo de Dijkstra si es aplicado sobre un grafo no conexo, resultará en un arreglo de distancias que siempre tendrá al menos un valor "infinito".