uniandes

Ingeniería de Sistemas y Computación

Diseño y análisis de algoritmos

Profesor: Mateo Sanabria Ardila

Preparación parcial 2

Fecha de entrega: * 2023-10 Nota máxima: *

---

1. Implemente todos los algoritmos vistos en clase y estudie su complejidad temporarl.
2. Explique cual es la complejidad temporal de los algoritmos vistos en clase: **BFS, Dijkstra, Bellman-Ford** y **Floyd-Warshall**.
3. Explique por que el algoritmo de Dijkstra no funciona en un grafo que tiene arcos negativos, de ser necesario puede usar un ejemplo para ilustrar su explicación.
4. Note que si se tiene un grafo de tamaño n se puede ejecutar **Dijkstra** o **Bellman-Ford** n veces y lograr el mismo resultado que **Floyd-Warshall**. Sabemos que se obtiene el mismo resultado, sin embargo, este acercamiento es equivalente? la complejidad temporal es la misma? la complejidad espacial?
5. Modifique el algoritmo de **Floyd-Warshall** de tal forma que sea posible construir el camino mas corto de entre cualquier nodo del grafo. Se modifica la complejidad espacial o temporal del algoritmo original?
6. Explique como el algoritmo de **Floyd-Warshall** puede detectar si el grafo original tiene ciclos negativos. Algún otro de los algoritmos visto en clase permite lo mismo?
7. Un grafo bipartito, es un grafo tal que el conjunto de nodos se puede expresar como dos conjuntos disjuntos, de manera que no hay dos nodos del mismo conjunto que sean adyacentes. Diseñe un algoritmo que reciba un grafo, en su implantación indique si su grafo esta implementado como lista o matriz de adyacencia, y retorne $True$ si el grafo es bipartito y $False$ en otro caso. La complejidad no debe superar $\mathcal{O}(|V||E|)$, donde V es el conjunto de nodos y E el conjunto de arcos del grafo.
8. Un grafo ponderado desconectado no tiene árboles de expansión. Sin embargo, es posible encontrar un bosque de expansión de peso mínimo en un grafo de este tipo. Explique cómo modificar tanto el algoritmo de Kruskal como el algoritmo de Prim para hacer esto.
9. Explique porque el flujo máximo se alcanza cuando la red residual no contiene caminos de aumento.

10. Describa un algoritmo eficiente que encuentre un nuevo flujo máximo si la capacidad de una arista concreta aumenta en una unidad. Describa un algoritmo eficiente que encuentre un nuevo flujo máximo si la capacidad de una arista concreta disminuye en una unidad.

11. Ejercicios propuestos de LeetCode:
    (a) 1334. Find the City With the Smallest Number of Neighbors at a Threshold Distance
    (b) 743. Network Delay Time
    (c) 1349. Maximum Students Taking Exam (Este es realmente un ejercicio difícil)