

Laboratorio 4: Algoritmos Voraces

Ricardo Rafael Azopardo Cadenas

Universidad Eafit
Medellín, Colombia
rrazopardc@eafit.edu.co

Jhon Jairo Chavarria Gaviria

Universidad Eafit
Medellín, Colombia
jjchavarrg@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Expliquen con sus propias palabras la estructura de datos que utiliza para resolver el problema del numeral 1.1 y cómo funciona el algoritmo.

El algoritmo utilizado para resolver el TSP en este caso es un algoritmo greedy, el vecino más cercano. El algoritmo recorre sobre un grafo representado por listas de adyacencia, el algoritmo del vecino más cercano en cada nodo que visita compara el camino más corto entre sus hijos (vecinos), marcando cada hijo para no volver a visitarlo, y regresa al nodo inicial. Es importante señalar que el grafo es completo, es decir, todos sus nodos están conectados con todos los nodos.

3.2 Para resolver el problema del agente viajero, usando un algoritmo voraz, aun cuando no arroje la solución óptima, ¿Qué debe cumplir el grafo para que el algoritmo, al menos, arroje una solución, así no sea óptima? ¿Por qué?

En el caso de la solución del numeral 1.1 solo funciona si de un nodo se pueden llegar a todos los otros nodos y además se debe poder volver al nodo inicial, de tal forma que es necesario que si se va a usar un algoritmo greedy, el grafo debe ser completo, si el grafo no es de esta manera pueden existir la posibilidad de nodos inalcanzables o se imposibilita volver al nodo inicial.

3.3 Expliquen con sus propias palabras la estructura de datos que utiliza para resolver el problema del numeral 2.1 y cómo funciona el algoritmo.

El algoritmo del numeral 2.1 tomando en consideración la especificación del problema funciona de la siguiente manera: se leen los datos iniciales: el número

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

de conductores/horarios de mañana y de tarde, el número d que se debe tratar de no superar y el r , que actúa como cantidad que se cobrará por cada hora extra que se acumule en un horario. Luego se leen ambos horarios de la mañana y la tarde en las siguientes dos líneas, se almacenan en dos arreglos separados y se ordenan los arreglos.

Luego se recorren ambos arreglos en un mismo ciclo, aplicando la siguiente fórmula para el horario en i en la mañana y el horario en $n-i-1$ en la tarde, es decir el primer horario de la mañana con el último de la tarde.

(Primeros Horarios Mañana + Últimos Horarios Tarde) – d

Esto se realiza de este modo debido a que como ambos arreglos están ordenados se estarían sumando los mayores términos de un arreglo con los menores del otro, por lo tanto se forman números globales muy parecidos y se evitan números muy grandes, se prosigue a restarles la cantidad d para comprobar que no se pasen a horas extra, o por el caso contrario encontrar una diferencia para multiplicar por r y acumular la suma mínima.

El algoritmo no encuentra la solución perfecta, pero si encuentra una solución global muy aproximada en un tiempo eficiente.

3.4 Calculen la complejidad del ejercicio trabajado en el numeral 2.1 y agréguela al informe PDF.

$O(n)$.

3.5 Expliquen con sus palabras las variables (qué es ' n ', qué es ' m ', etc.) del cálculo de complejidad del numeral anterior.

N hace referencia al número de operaciones que se deben repetir de acuerdo a esta cantidad, en este caso como solo involucra un ciclo a la vez, logra permanecer en esta complejidad.

4) Simulacro de Parcial

1. $i = j$;
2. $\min > \text{adjacencyMatrix}[\text{element}][i]$

3. a.

Paso	Visitados	B	C	D	E	F	G	H
1	A	20,A	∞	80,A	∞	∞	90,A	∞
2	B	20,A	∞	80,A	∞	30,B	90,A	∞
3	F	20,A	40,F	70,F	∞	30,B	90,A	∞
4	C	20,A	40,F	50,C	∞	30,B	90,A	60.C
5	D	20,A	40,F	50,C	∞	30,B	70,D	60.C
6	H	20,A	40,F	50,C	∞	30,B	70,D	60.C
7	G	20,A	40,F	50,C	∞	30,B	70,D	60.C

3.b $A \rightarrow B \rightarrow F \rightarrow C \rightarrow D \rightarrow G$