

	<p style="text-align: center;">UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</p>	Código: ST245
		Estructura de Datos 2

Laboratorio Nro. 4: Algoritmos voraces

Juan Camilo Arenas
Universidad Eafit
Medellín, Colombia
Jarena12@eafit.edu.co

Nombre completo de integrante 2
Universidad Eafit
Medellín, Colombia
Correointegrante2@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

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:

RESPUESTA:

Lee en consola los casos de prueba y las rutas de día y noche hasta que el usuario ingrese "0 0 0" o llegue a 100 casos de prueba.
Almacena en la variable resultados los resultados de cada caso de prueba,
Luego
Imprime en consola el arreglo con los resultados de los casos de prueba
Y
Procesa cada caso de prueba.
Recibe la cadena caso de prueba y las rutas de día y de noche.
Retorna el minimo valor posible de horas extras para el caso de prueba.
Convierte las cadenas ingresadas en arrays
Y
Para cada n suma los valores de las rutas de día y noche,
si el resultado es mayor a minimoValor se asigna como nuevo minimoValor
Y por ultimo
/Retorna minimoValor que es el acumulado menos d (maximo horas por día) y lo multiplica por r (valor hora)

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

$O(k)$

Donde k es el número de rutas en la mañana y la tarde.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

3) Simulacro de Parcial en el informe PDF

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

PASO	A	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, D	∞	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
8	E	20, A	40, F	50, C	∞	30, B	70, D	60, C

B) $A \rightarrow B \rightarrow F \rightarrow C \rightarrow D \rightarrow G$

- 4.) a) temp /
- 2 b) temp + minimo
- c) b

RESUMEN DE LECTURA.

Algorithms Avidos.

La sección 10.3 nos empieza a hablar sobre el ejemplo de las monedas conocida por los algoritmos voraces nos dice que supongamos que tenemos que dar un cambio suponiendo monedas de 25c, 10c, 5c, y 1c, y que deseo dar un cambio de 63c. tenemos que determinar una lista de monedas con el valor correcto, si no que se produjo la lista mas corta de monedas con ese valor.

Este método de dar cambio es un algoritmo ávido. En cualquier etapa individual, un algoritmo ávido selecciona la opción que sea localmente optima, y nos dice que observemos que el algoritmo ávido para dar cambio produce una solución óptima general debido a las propiedades de la moneda.

Es necesario subrayar que el hecho de que no todos los enfoques ávidos llegan a dar mejor resultado. Como sucede en la vida real, una estrategia ávida puede dar un buen resultado durante un tiempo, pero el resultado general puede ser pobre. Como ejemplo, se considera lo que sucede al permitir aristas ponderadas negativas en los algoritmos de Dijkstra y de Kruskal.