Laboratorio Nro. 4: Algoritmos voraces

Luisa María Vásquez Gómez
Universidad Eafit
Medellín, Colombia
lmvasquezg@eafit.edu.co

Juan José Parra Díaz
Universidad Eafit
Medellín, Colombia
jjparrad@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1. El algoritmo del punto 1 funciona buscando en el nodo de inicio el sucesor más cercano a este ,al hallarlo, este es agregado al camino y se realiza simultáneamente el mismo proceso moviéndose a través del grafo en aquellos vértices que no han sido visitados; una vez recorrido todo el grafo, se busca un camino hacia el inicio de nuevo, si este es sucesor inmediato del ultimo vértice del camino, simplemente vuelve al origen, de lo contrario, busca un camino a este con el mismo método usado anteriormente, eso sí, con el costo de que tendría que visitar ciudades ya visitadas, todo esto depende de cómo este estructurado el grafo.

2. Para poder dar solución al problema del agente viajero es indispensable que exista al menos un arco que llegue al origen, ya que si no lo hay, se vuelve imposible encontrar un camino que termine en este; además, todas las ciudades deben tener arcos de entrada y salida, ya que a falta de estos se puede llegar a un estado en el que no se pueden visitar mas ciudades, o la ciudad en cuestión nunca podría ser visitada.

3. El ejercicio en línea 2.1 funciona leyendo por medio de un BufferedReader, la información de la entrada, la cual, al finalizar la lectura de cada caso, guarda todos los datos en un objeto nuevo de tipo “Bus” (objeto el cual contiene tres enteros para tener la información de la empresa de buses, y dos arreglos de enteros que guardan los horarios de la mañana y de la tarde respectivamente). Los casos son guardados en un ArrayList de Buses para de esta manera, manejar cada uno de los casos por aparte.

   Cuando ya está creada la estructura de datos, se procede al algoritmo voraz que define el valor de las horas mínimas a pagar, sumando las horas de cada recorrido matutino y vespertino, y restándolas del máximo número de horas por día. El restante lo multiplica por el valor de la hora extra. Se retorna el

valor total y se imprime. Se continúa con el siguiente caso hasta finalizarlos todos.

4.

```
public void inputManager() throws IOException{                    O(n*r)
    buses = new ArrayList<>();                                          C

    String[] datos;                                                     C
    int[] temprano;                                                     C
    int[] tarde;                                                        C

    BufferedReader br = new BufferedReader(new                          C
InputStreamReader(System.in));
    String entrada = br.readLine();                                     C
    while(!entrada.equals("0 0 0")){                                    C*n
        datos = entrada.split(" ");                                     C*n
        int r = Integer.parseInt(datos[0]);                             C*n
        int m = Integer.parseInt(datos[1]);                             C*n
        int e = Integer.parseInt(datos[2]);                             C*n

        entrada = br.readLine();                                        C*n
        datos = entrada.split(" ");                                     C*n
        temprano = new int[datos.length];                               C*n
        for(int i = 0; i < r; i++){                                     C*n*r
            temprano[i] = Integer.parseInt(datos[i]);                   C*n*r
        }

        entrada = br.readLine();                                        C*n
        datos = entrada.split(" ");                                     C*n
        tarde = new int[datos.length];                                  C*n
        for(int i = 0; i < r; i++){                                     C*n*r
            tarde[i] = Integer.parseInt(datos[i]);                      C*n*r
        }
```

```
            buses.add(new Bus(r,m,e,temprano,tarde));                    C*n

            entrada = br.readLine();                                     C*n
        }
    }

    public void getExtra(){                                              O(n*r)
        for(Bus b: buses){                                               C*n
            System.out.println(b.pagoExtra());                           O(r)*n
        }
    }

    public int pagoExtra(){                                              O(r)
        int extraTotal = 0;
        for(int i = 0; i < rutas; i++){                                  C*r
            int horasTotal = temprano[i] + tarde[i];                     C*r
            int horasExtra = horasTotal - horasMax;                      C*r
            if(horasExtra > 0){                                          C*r
                extraTotal += valorHorasExtra*horasExtra;                C*r
            }
        }
        return extraTotal;                                               C
    }

    public static void main(String[] args){                              O(n*r)
        EjercicioEnLinea eel = new EjercicioEnLinea();                   C
        System.out.println();                                            C
        try{
            eel.inputManager();                                          O(n*r)
        } catch(IOException e){}
        eel.getExtra();                                                  O(n)
    }
```

Complejidad: O(n*r)

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

5. En el cálculo de la complejidad en el punto anterior hay 2 variables: ‘r’ y ‘n’. La variable ‘n’ representa el número de casos a los que se les tenga que aplicar el algoritmo, mientras que la variable ‘r’ es el número más alto de rutas que haya en cualquiera de los casos.

4) Simulacro de parcial LAB 4

1. i = j;

2. min > adjacencyMatrix [element] [i]

3. **a)**

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,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
8	E	20,A	40,F	50,C	∞	30,B	70,D	60,C

b) A->B->F->C->D->G

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