AYED II

Lautaro Gastón Peralta

Práctico 3.2

1. Ejecutar paso a paso, graficando las soluciones parciales, el algoritmo de Prim que computa el *árbol generador mínimo* sobre los grafos con nodos $\{1, 2, \ldots, 8\}$ y costos dados por una función w:

(a)

$$\begin{array}{llll} w((1,2)) = 7 & w((2,3)) = 4 & w((3,6)) = 4 & w((5,6)) = 6 \\ w((1,6)) = 3 & w((2,4)) = 2 & w((3,8)) = 6 & w((6,7)) = 5 \\ w((1,7)) = 5 & w((2,5)) = 1 & w((4,6)) = 8 & w((8,5)) = 2 \\ w((1,3)) = 3 & w((3,4)) = 5 & w((5,4)) = 3 & w((8,7)) = 3 \end{array}$$

(b)

$$\begin{array}{llll} w((1,2)) = 3 & w((2,3)) = 1 & w((3,6)) = 3 & w((5,6)) = 6 \\ w((1,6)) = 2 & w((2,4)) = 1 & w((3,8)) = 7 & w((6,7)) = 6 \\ w((1,7)) = 8 & w((2,5)) = 5 & w((4,6)) = 1 & w((8,5)) = 1 \\ w((1,3)) = 1 & w((3,4)) = 9 & w((5,4)) = 2 & w((8,7)) = 5 \end{array}$$

a)

paso 0 : C={1,2,3,4,5,6,7,8}. T={}

paso 1 : elimino el vértice inicial, k=1. C={2,3,4,5,6,7,8}. T={}

paso 2 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones : {(1,2)[7];(1,6)[3];(1,7)[5];(1,3)[3]}

selecciono (1,3) con costo 3, v1=1 ; v2 = 3

como v2 pertenece a C, elimino a v2 del conjunto C. C={2,4,5,6,7,8}. T={(1,3)}

paso 3 :

c:= selecciono arista de costo mínimo tq conecta un vertice en C con uno fuera de C.

opciones : {(1,2)[7];(1,6)[3];(1,7)[5];(2,3)[4];(3,4)[5];(3,6)[4];(3,8)[6]}

selecciono (1,6) con costo 3, v1=1 ; v2 = 6

como v2 pertenece a C, elimino a v2 del conjunto C. C={2,4,5,7,8}. T={(1,3);(1,6)}

paso 4 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones : {(1,2)[7];(1,7)[5];(2,3)[4];(3,4)[5];(3,6)[4];(3,8)[6];(4,6)[8],(5,6)[6],(6,7)[5]}

selecciono (2,3) con costo 4, v1=2 ; v2 = 3

como v1 pertenece a C, elimino a v1 del conjunto C. C={4,5,7,8}. T={(1,3);(1,6);(2,3)}

paso 5 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones : {(1,2)[7];(1,7)[5];(3,4)[5];(3,6)[4];(3,8)[6];(4,6)[8],(5,6)[6],(6,7)[5];(2,4)[2];(2,5)[1]}

selecciono (2,5) con costo 1, v1=2 ; v2 = 5

como v2 pertenece a C, elimino a v2 del conjunto C. C={4,7,8}. T={(1,3);(1,6);(2,3);(2,5)}

paso 6 :

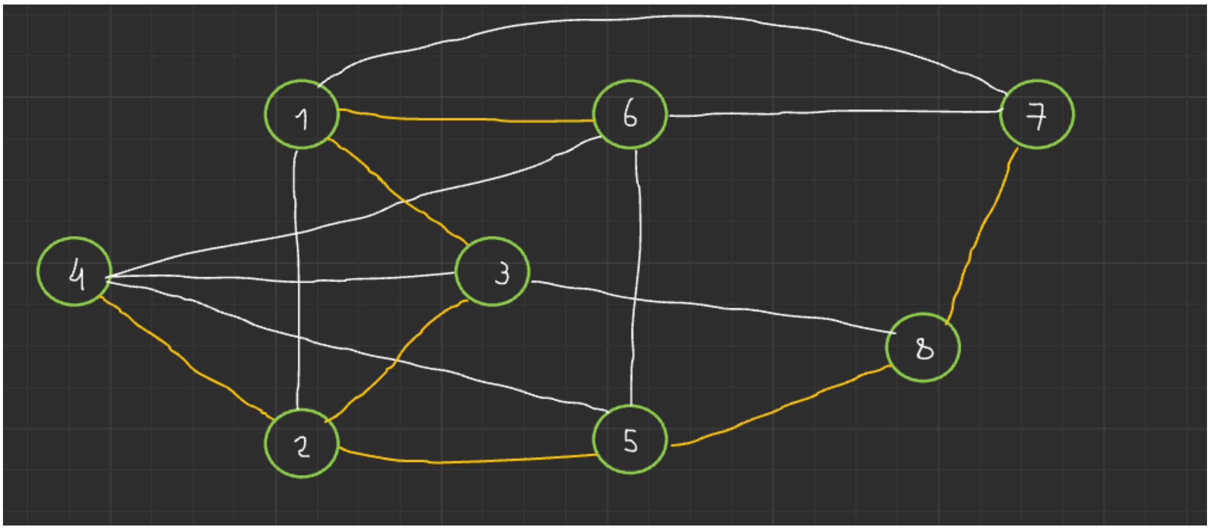
c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones :
{(1,2)[7];(1,7)[5];(3,4)[5];(3,6)[4];(3,8)[6];(4,6)[8],(5,6)[6],(6,7)[5];(2,4)[2];(5,4)[3],(8,5)[2]}
selecciono (2,4) con costo 1, v1=2 ; v2 = 4
como v2 pertenece a C, elimino a v2 del conjunto C. C={7,8}.
T={(1,3);(1,6);(2,3);(2,5);(2,4)}

paso 7 :
c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.
opciones :
{(1,2)[7];(1,7)[5];(3,4)[5];(3,6)[4];(3,8)[6];(4,6)[8],(5,6)[6],(6,7)[5];(5,4)[3];(8,5)[2]}
selecciono (8,5) con costo 1, v1=8 ; v2 = 5
como v1 pertenece a C, elimino a v1 del conjunto C. C={7}.
T={(1,3);(1,6);(2,3);(2,5);(2,4);(8,5)}

paso 8 :
c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.
opciones :
{(1,2)[7];(1,7)[5];(3,4)[5];(3,6)[4];(3,8)[6];(4,6)[8],(5,6)[6],(6,7)[5];(5,4)[3];(8,7)[3]}
selecciono (8,7) con costo 1, v1=8 ; v2 = 7

como v2 pertenece a C, elimino a v2 del conjunto C. C={}.
T={(1,3);(1,6);(2,3);(2,5);(2,4);(8,5);(8,7)}



b)
paso 0 : C={1,2,3,4,5,6,7,8}. T={}
paso 1 : elimino el vértice inicial, k=1. C={2,3,4,5,6,7,8}. T={}
paso 2 :
c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.
opciones : {(1,2)[3];(1,6)[2];(1,7)[8];(1,3)[1]}
selecciono (1,3) con costo 1, v1=1 ; v2 = 3
como v2 pertenece a C, elimino a v2 del conjunto C. C={2,4,5,6,7,8}. T={(1,3)}
paso 3 :

c:= selecciono arista de costo mínimo tq conecta un vertice en C con uno fuera de C.

opciones : {(1,2)[3];(1,6)[2];(1,7)[8];(2,3)[1];(3,4)[9];(3,6)[3];(3,8)[7]}

selecciono (2,3) con costo 1, v1=2 ; v2 = 3

como v2 pertenece a C, elimino a v2 del conjunto C. C={4,5,6,7,8}. T={(1,3);(2,3)}

paso 4 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones : {(1,2)[3];(1,6)[2];(1,7)[8];(3,4)[9];(3,6)[3];(3,8)[7];(2,4)[1];(2,5)[5]}

selecciono (2,4) con costo 1, v1=2 ; v2 = 4

como v2 pertenece a C, elimino a v2 del conjunto C. C={5,6,7,8}. T={(1,3);(2,3);(2,4)}

paso 5 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones : {(1,2)[3];(1,6)[2];(1,7)[8];(3,4)[9];(3,6)[3];(3,8)[7];(2,5)[5];(4,6)[1];(5,4)[2]}

selecciono (4,6) con costo 1, v1=4 ; v2 =6

como v2 pertenece a C, elimino a v2 del conjunto C. C={5,7,8}.

T={(1,3);(2,3);(2,4);(4,6)}

paso 6 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones :

{(1,2)[3];(1,6)[2];**(1,7)[8]**;(3,4)[9];(3,6)[3];**(3,8)[7]**;**(2,5)[5]**;**(5,4)[2]**;**(5,6)[6]**;**(6,7)[6]**}

selecciono (5,4) con costo 2, v1=5 ; v2 = 4

como v1 pertenece a C, elimino a v1 del conjunto C. C={7,8}.

T={(1,3);(2,3);(2,4);(4,6);(5,4)}

paso 7 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

opciones :

{(1,2)[3];(1,6)[2];**(1,7)[8]**;(3,4)[9];(3,6)[3];**(3,8)[7]**;**(2,5)[5]**;**(5,6)[6]**;**(6,7)[6]**;**(8,5)[1]**}

selecciono (8,5) con costo 1, v1=8 ; v2 = 5

como v1 pertenece a C, elimino a v1 del conjunto C. C={7}.

T={(1,3);(2,3);(2,4);(4,6);(5,4);(8,5)}

paso 8 :

c:= selecciono arista de costo mínimo tq conecta un vértice en C con uno fuera de C.

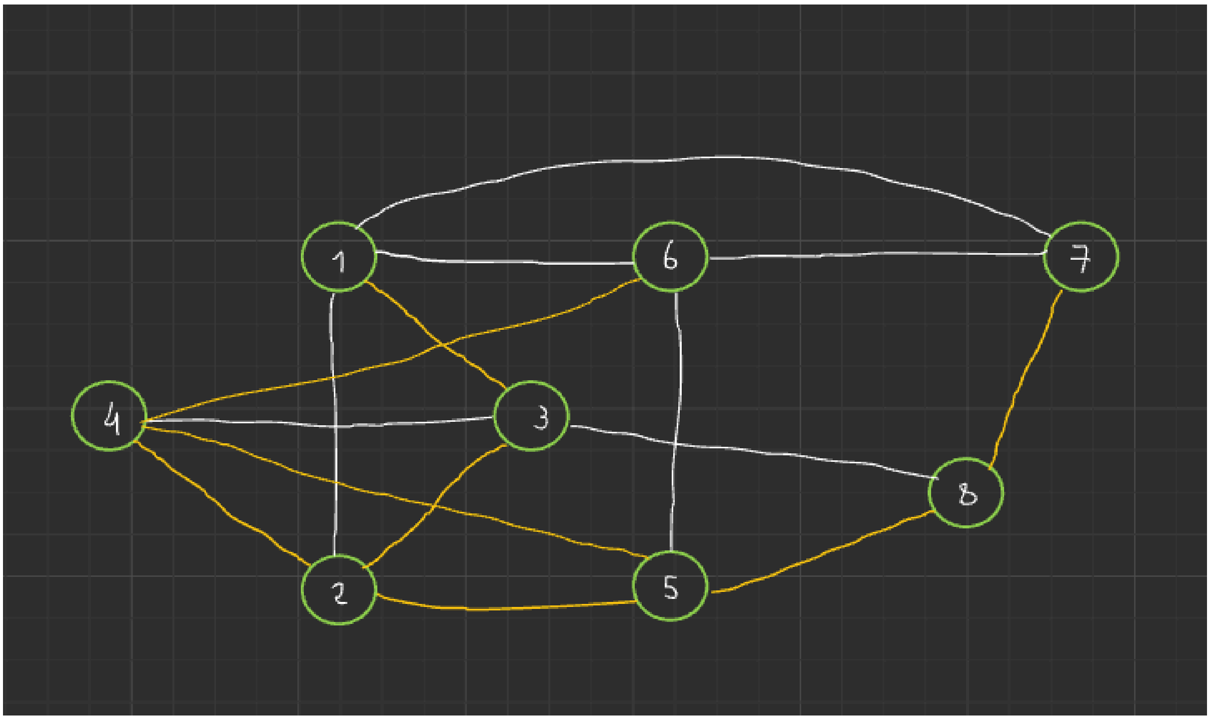
opciones :

{(1,2)[3];(1,6)[2];**(1,7)[8]**;(3,4)[9];(3,6)[3];(3,8)[7];(2,5)[5];(5,6)[6];**(6,7)[6];(8,7)[5]**}

selecciono (8,7) con costo 1, v1=8 ; v2 = 7

como v2 pertenece a C, elimino a v2 del conjunto C. C={}.

T={(1,3);(2,3);(2,4);(4,6);(5,4);(8,5);(8,7)}



2. Ejecutar paso a paso el algoritmo de Dijkstra que computa el *camino de costo mínimo* entre un nodo dado y los restantes nodos de un grafo, sobre los dos grafos especificados en el ejercicio anterior.

   Considerar 1 como el nodo inicial. Explicitar en cada paso el conjunto de nodos para los cuales ya se ha computado el costo mínimo y el arreglo con tales costos.

a)

C = {} D:=[] v := 1

C = {1,2,3,4,5,6,7,8} D:=[]

C = {2,3,4,5,6,7,8} D:=[0,7,3,infinito,infinito,3,5,infinito]

{-CICLO-}

iter1 : C = {2,3,4,5,6,7,8} D:=[0,7,3,infinito,infinito,**3**,5,infinito] **c:=6**

C = {2,3,4,5,7,8} D:=[0,7,3,11,9,**3**,5,infinito]

iter2 : C = {2,3,4,5,7,8} D:=[0,7,3,11,9,**3**,5,infinito] **c:=3**

C = {2,4,5,7,8} D:=[0,7,**3**,8,9,**3**,5,9]

iter3 : C = {2,4,5,7,8} D:=[0,7,**3**,8,9,**3**,5,9] **c:=7**

C = {2,4,5,8} D:=[0,7,**3**,8,9,**3**,**5**,8]

iter4 : C = {2,4,5,8} D:=[0,7,**3**,8,9,**3**,**5**,8] **c:=2**

C = {4,5,8} D:=[0,**7**,**3**,8,8,**3**,**5**,8]

iter5 : C = {4,5,8} D:=[0,**7**,**3**,8,8,**3**,**5**,8] **c:=4**

C = {5,8} D:=[0,**7**,**3**,**8**,8,**3**,**5**,8]

iter6 : C = {5,8} D:=[0,**7**,**3**,**8**,8,**3**,**5**,8] **c:=5**

C = {8} D:=[0,**7**,**3**,**8**,**8**,**3**,**5**,8]

iter7 : C = {8} D:=[0,**7**,**3**,**8**,**8**,**3**,**5**,8] **c:=8**

C = {} D:=[0,**7**,**3**,**8**,**8**,**3**,**5**,**8**]

b)

C = {} D:= [] v:= 1

C = {1,2,3,4,5,6,7,8} D:= []

C = {2,3,4,5,6,7,8} D:=[0,3,1,infinito,infinito,2,8,infinito]

{-CICLO-}

iter1 : C = {2,3,4,5,6,7,8} D:=[0,3,**1**,infinito,infinito,2,8,infinito] **c:=3**

C = {2,4,5,6,7,8} D:=[0,1,**1**,10,infinito,2,8,8]

iter2 : C = {2,4,5,6,7,8} D:=[0,1,**1**,10,infinito,2,8,8] **c:=6**

C = {2,4,5,7,8} D:=[0,1,**1**,3,8,**2**,8,8]

iter3 : C = {2,4,5,7,8} D:=[0,1,**1**,3,8,**2**,8,8] **c:=2**

C = {4,5,7,8} D:=[0,**1**,**1**,2,6,**2**,8,8]

iter4 : C = {4,5,7,8} D:=[0,**1**,**1**,2,6,**2**,8,8] **c:=4**

C = {5,7,8} D:=[0,**1**,**1**,**2**,4,**2**,8,8]

iter5 : C = {5,7,8} D:=[0,**1**,**1**,**2**,4,**2**,8,8] **c:=5**

C = {7,8} D:=[0,**1**,**1**,**2**,**4**,**2**,8,8]

iter6 : C = {7,8} D:=[0,**1**,**1**,**2**,**4**,**2**,8,8] **c:=7**

C = {8} D:=[0,**1**,**1**,**2**,**4**,**2**,**8**,8]

iter6 : C = {7,8} D:=[0,**1**,**1**,**2**,**4**,**2**,8,8] **c:=8**

C = {} D:=[0,**1**,**1**,**2**,**4**,**2**,**8**,**8**]

3. Usted quiere irse de vacaciones y debe elegir una ciudad entre K posibles que le interesan. Como no dispone de mucho dinero, desea que el viaje de ida hacia la ciudad pueda realizarse con a lo sumo L litros de nafta.

   (a) Dé un algoritmo que, dado un grafo representado por una matriz E : **array**[1..n,1..n] **of** Nat, donde el elemento E[i,j] indica el costo en litros de nafta necesario para ir desde la ciudad i hasta la ciudad j; un conjunto C de vértices entre 1 y n, representando las ciudades que quieren visitarse; un vértice v, representando la ciudad de origen del viaje; y un natural L, indicando la cantidad de litros de nafta total que puede gastar; devuelva un conjunto D de aquellos vértices de C que puede visitar con los L litros.

   (b) Ejecute el algoritmo implementado en el inciso anterior para el grafo descripto en el siguiente gráfico, con vértices 1,2,...11, tomando C = {11,5,10,7,8} como las ciudades de interés, disponiendo de L = 40 litros de nafta. ¿Cuáles son los posibles destinos de acuerdo a su presupuesto?

   Ayuda: Utilice el algoritmo de Dijkstra.