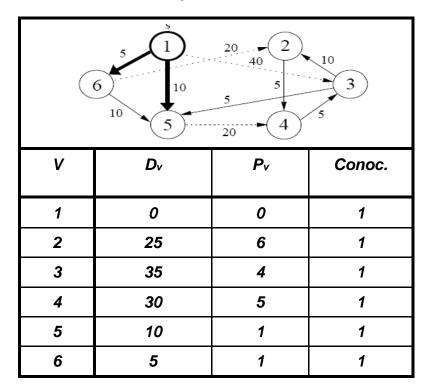
Algoritmos y Estructuras de Datos - Redictado 2019 -

Ejercitación sobre Grafos – Caminos mínimos: Algoritmo de Dijkstra. Variante para pesos negativos

Algoritmo de Dijkstra

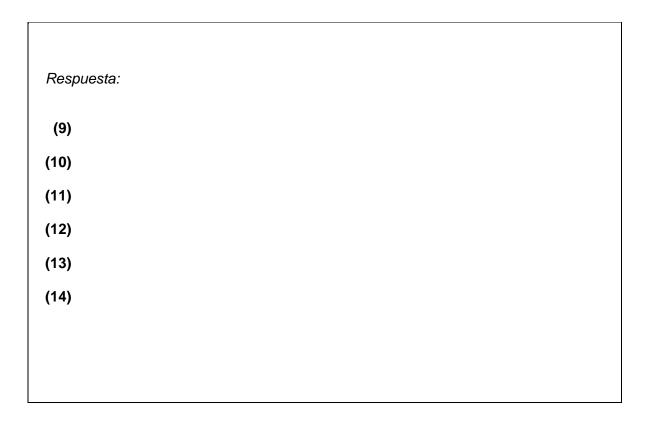
1.- Dado el siguiente grafo, recuperar los vértices de los caminos de costo mínimo obtenidos con el algoritmo de Dijkstra, para los siguientes pares (origen,destino):

Recuperar caminos

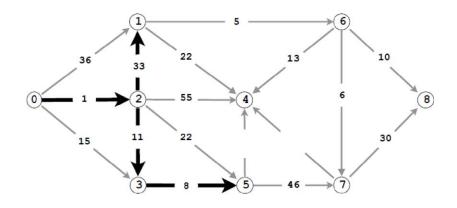


2.- a.- Complete el siguiente seudo-código del algoritmo de Dijkstra:

```
Dijkstra(G,w, s)
(1)
               para cada vértice v EV {
(2)
                      Dv = \infty; Pv = 0;
(3)
(4)
               Ds = 0;
               para cada vértice v EV {
(5)
                 u = vérticeDesconocidoDeMenorDistancia de vector D;
(6)
                Marcar u como conocido;
(7)
                para cada vértice w E V adyacente a u {
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
                }
```



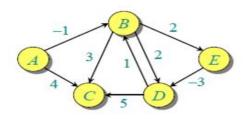
- b.- (1) ¿Cuál es el Orden de ejecución de esta versión del algoritmo?.
 - (2) ¿Puede mejorarse?
 - (3) ¿Cómo?
- 3.- Dado el grafo de la figura:



- a) Teniendo en cuenta que puede haber más de un camino mínimo entre el origen y los restantes vértices, modificar el algoritmo de Dijkstra para contar la cantidad de caminos mínimos para cada vértice. Por ejemplo: 2 caminos de costo mínimo 52 entre 0 y 4; 1 camino de costo mínimo 34 entre 0 y 1.
- b) ¿Qué observa al calcular el camino de costo mínimo entre 0 y 4?

Grafos con pesos positivos y negativos

4.- a) Aplicar el algoritmo de Dijkstra al siguiente grafo con pesos positivos y negativos. Observar que Dijkstra no siempre funciona bien.



Orden en que se toma el vértice	Vértices "v"	Distancia_a (w)	Vértice "w"	Visitado
10	A	0		1
	В	ω		0
	С	ω		0
	D	∞		0
	E	ω		0

a) Resolverlo aplicando la variante de Dijkstra que usa una Cola

```
Camino_min_GrafoPesosPositivosyNegativosG,s)
(1)
       D_s = 0; Encolar (Q,s);
       Mientras (not esVacio(Q)) {
(2)
         Desencolar(Q,u);
(3)
(4)
         para c/vértice w € V adyacente a u {
               si (D_w > D_u + c(u, w)) {
(5)
(6)
                       D_w = D_u + c(u, w);
(7)
                       P_w = u;
                      si (w no está en Q)
(8)
                         Encolar(Q,w);
(9)
(10)
               }
          }
(11)
(12)
       }
       }
```

Orden en que se toma el vértice	Vértices "v"	Distancia_a (w)	Vértice "w"	Visitado
1º	Α	0		1
	В	∞		0
	С	∞		0
	D	∞		0
	E	∞		0