Práctico 3 ALGABO: El algoritmo de Dijkstra de caminos más cortos con pesos no negativos.

Mauricio Velasco

1. Sea G un grafo dirigido con pesos positivos y sea t un vértice de G:

   a) Proponga una variante del algoritmo de Dijkstra para calcular las distancias **hasta** t, es decir los valores $d_\ell(v,t)$ para $v \in V(G)$.

   b) Demuestre que su algoritmo es correcto.

2. * Sea G un grafo dirigido con pesos no negativos. Defina el costo-bottleneck de un camino P como

   $$b(P) = \text{Máxima longitud } \ell \text{ de una arista de } P$$

   y para dos vértices u y v defina la onstrucciones bottleneck $m(s,v)$ como

   $$m(s,v) = \min\{b(P) : P \text{ es un camino de } s \text{ a } v \text{ en } G\}$$

   a) Calcule $m(s,v)$ para todo $v \in V$ para el grafo $[M3]$ de clase.

   b) Proponga una variante del algoritmo de Dijkstra para calcular obstrucciones bottleneck en tiempo $O(mn)$.

   c) Demuestre la validez del algoritmo que propone.

3. Implemente el algoritmo de Dijkstra en python dos veces, una vez usando un `priority_queue` o `heap` (por ejemplo usando `import heapq`) y la segunda vez sin usar esta estructura de datos especial.

   a) Escriba el código de sus dos implementaciones.

   b) Ejecute sus algoritmos en el ejemplo $[M3]$ y escriba en qué orden se incluyen los vértices en el conjunto X.

   c) Ejecute sus implementaciones en ejemplos construidos por uds. de grafos grandes y lleve sus implementaciones hasta el límite. Es cierto en la práctica que usar el heap es mejor que no usarlo? Explique qué ejemplos de grafos utilizó y escriba una tabla con sus conclusiones.

4. Verdadero ó Falso? En el primer caso escriba una demostración y en el segundo encuentre un contrajemplo: *Suponga que el algoritmo de Dijkstra iniciando en* s *inserta los vértices de* $V(G)$ *en* X *en orden* $s = v_0, v_1, \ldots, v_n$. *Si* $i < j$ *entonces* $d_\ell(s, v_i) \leq d_\ell(s, v_j)$.