Examen 4

Análisis y Diseño de Algoritmos

8 de julio de 2022

Indicaciones específicas:

- Duración: 120 minutos
- El examen consiste de 10 preguntas. Debe entregar preguntas que en su total, como máximo, sumen 20 puntos. De no cumplir este requisito su nota será cero.
- **NO** se permite el uso de calculadoras, copias, apuntes ni libros

Ejercicio 1 (2 ptos). Sea G un grafo dirigido con pesos w en las aristas. Sea h una función que asigna valores reales a los vértices de G. Sea w' la función de pesos resultante de adicionar el valor $h(u)-h(v)$ en cada arista uv de G. Es decir, $w'(uv) = w(uv)+h(u)-h(v)$. Demuestre la siguiente propiedad. Si P es un camino de s a t en G, entonces $w'(P) = w(P)+h(s)-h(t)$.

Ejercicio 2 (2 ptos). Demuestre la siguiente propiedad. Sea G un grafo dirigido con longitudes ℓ en las aristas y tres vértices u, v, w. Entonces

$$dist(u, v) + dist(v, w) \geq dist(u, w)$$

donde $dist(x, y)$ guarda la distancia de x hacia y.

Ejercicio 3 (2 ptos). Sugiera un cambio para el algoritmo de Bellman-Ford para que termine antes de las $n-1$ iteraciones en caso sea posible. Dicho cambio no debe afectar el tiempo de ejecución del algoritmo.

Ejercicio 4 (3 ptos). Describa un algoritmo que, dados n enteros en el rango 0 a k, preprocesa su entrada en $\Theta(n + k)$ y luego responde en $O(1)$ cuantos de los n elementos están en un rango $[a, b]$ dado como query.

Ejercicio 5 (3 ptos). Respecto al algoritmo Bucketsort. ¿Cual es el número esperado de elementos ubicados en el i-ésimo bucket? Justifique adecuadamente.

Ejercicio 6 (3 ptos). Indique como ordenar n números enterios en el rango de 0 a $n^3 - 1$ en $O(n)$ **usando radix sort**. Basta dar una idea bien definida y un ejemplo concreto.

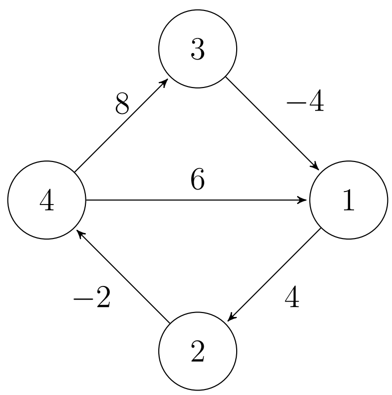
Ejercicio 7 (3 ptos). Respecto al algoritmo de Floyd-Warshall:

Floyd-Warshall(W)

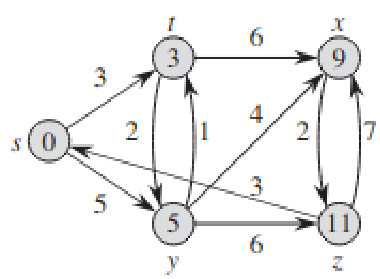
$M^{(0)} \leftarrow W$
for $k = 1$ hasta n
    for $i = 1$ hasta n
        for $j = 1$ hasta n
            $M_{ij}^{(k)} \leftarrow \text{mín}\{M_{ij}^{(k-1)}, M_{ik}^{(k-1)} + M_{kj}^{(k-1)}\}$
return $M^{(n)}$

Demuestre por inducción que $M_{ii}^{(k)} = 0$ para todos $1 \leq i, k \leq n$.

Ejercicio 8 (3 ptos). Ejecute el algoritmo $O(n^3 \lg n)$ visto en clase para encontrar caminos mínimos entre todos los pares en el siguiente grafo.



Ejercicio 9 (3 ptos). Muestre el valor del vector **dist** en cada iteración del algoritmo de Dijkstra, a partir del nodo s, y suponiendo que los vértices son procesados en orden alfabético. Muestre también el árbol de parentezco generado.



Ejercicio 10 (4 ptos). Considere la siguiente operación sobre un arreglo $A[1..n]$ de booleanos.

Delete(A)

$i \leftarrow p$
while $i \leq n$ AND $A[i] ==$ False
    $i \leftarrow i + 1$
$p \leftarrow i$
$(p \leq n)$? $(A[p] \leftarrow$ False$)$: (Escribir mensaje)

Considere que p es una variable global. También puede pensar que p es un atributo de una estructura con un arreglo y una operación llamada Delete. Considere una secuencia de n operaciones que comienza con un arreglo booleano inicializado en TRUE y con $p = 1$. Sean $p_0, p_1, \ldots, p_n$ los respectivos valores de p a lo largo de estas n ejecuciones. Sea c_i el costo de la i-ésima operación.

(*a*) Exprese c_i en función de los valores de p. Considere que cada línea de código cuesta 1.

(*b*) Aplique el método agregado para demostrar que el tiempo amortizado de la operación es $O(1)$.

(*c*) Considere los siguientes costos ficticios $\hat{c}$.

$$\hat{c_i} = \begin{cases} 2n + 4 & : i = 1 \\ 4 & : \text{caso contrario} \end{cases}$$

Aplique el método de recargas para demostrar que el tiempo amortizado de la operación es $O(1)$. Debe **demostrar por inducción como visto en clase.**