Comenzado el	martes, 15 de junio de 2021, 19:52
Estado	Finalizado
Finalizado en	martes, 15 de junio de 2021, 20:22
Tiempo empleado	30 minutos
Calificación	**9** de 10 (**90**%)

Pregunta **1**
Correcta
Puntúa 1 sobre 1

¿Cuáles de las siguientes son ciertas en relacion al algoritmo de Karger que permite obtener el corte mínimo de un grafo G con *n* vértices y *m* arcos? (Observación: más de una puede válida, por lo que marque TODAS las que considere correctas).

Seleccione una o más de una:

- [x] a. Es un algoritmo de base aleatoria tal que si se hacen $T = n^2$ repeticiones tiene una probabilidad de falla menor o igual que $1 / e$. ✔
- [x] b. Es un algoritmo de base aleatoria tal que si se hacen $T = n^2 * ln(n)$ repeticiones tiene una probabilidad de falla menor o igual que $1 / n$. ✔
- [x] c. Es un algoritmo de base aleatoria cuya probabilidad de falla en todos los intentos *P(FT)* viene dada por la expresión $P(FT) <= (1 - 1/n^2)^T$ ✔
- [ ] d. Es un algoritmo de base aleatoria tal que si se hacen $T = n^2 * ln(n)$ repeticiones tiene una probabilidad de falla menor o igual que $1 / n^2$.

Pregunta **2**
Correcta
Puntúa 1 sobre 1

Suponga que se está trabajando en el análisis de ciertas propiedades numéricas, y se le pide crear un grafo dirigido para modelar la siguiente relación para los números enteros en el intervalo [1..10]:

```
x R y =>  (x < y) && (x * y < 20)   [con x, y en [1..10]]
```

Para aclarar: los vértices del grafo serán los números del 1 al 10, y la relación *x R y* pide que se establezca un arco que parta de x y llegue a *y* siempre que se cumpla que *x* es menor que *y* y que el producto *x * y* sea menor a 20.

¿Cuáles de las siguientes son ciertas en relacion al grafo que surge de aplicar esta relación? (Observación: más de una puede válida, por lo que marque TODAS las que considere correctas).

Seleccione una o más de una:

- [x] a. Existe al menos un vértice en este grafo, que no tiene arcos de llegada. ✔
- [ ] b. Todos los vértices del grafo sirven como vértice de partida para al menos un arco.
- [ ] c. El grafo es denso.
- [x] d. El grafo no tiene ningún autociclo. ✔

Pregunta **3**

Correcta

Puntúa 1 sobre 1

Sea un grafo **dirigido** *G = {V, A}* donde *V* es el conjunto de *n* vértices y *A* es el conjunto de *m* arcos. Se define como el ***entre grado*** *de un vértice x* del grafo *G* a la cantidad de arcos *que tienen a x como vértice de llegada*. Considere tanto la implementación matricial del grafo como la implementación por listas de adyacencia que se sugirió en el curso. ¿Cuántas operaciones en el peor caso requerirá realizar un método que calcule el *entre grado* de un vértice *x* cualquiera, en ambas implementaciones? (Observación: se supone que *x* es el objeto que se almacena en el vértice [y por lo tanto, en ambas implementaciones debe primero buscarlo en el vector de nodos o en la lista de vértices]. Recuerde que el grafo se supone ***dirigido***).

Seleccione una:

a.
- *Implementación matricial*: *n* comparaciones para buscar *x* en el vector de nodos y otras *n* en la columna de *x* en la matriz de adyacencias para contar los arcos que llegan a *x*. ✔
- *Implementación encadenada*: *n* comparaciones para buscar *x* en la lista de vértices, y otras *m* para recorrer la lista de arcos de *x* y contar aquellos que tienen a *x* como vértice final.

b.
- *Implementación matricial*: *n* comparaciones para buscar *x* en el vector de nodos y otras *n* en la columna de *x* en la matriz de adyacencias para contar los arcos que llegan a *x*.
- *Implementación encadenada*: *n* comparaciones para buscar *x* en la lista de vértices, y otras m^2 para recorrer la lista de arcos de *x* y contar aquellos que tienen a *x* como vértice final.

c.
- *Implementación matricial*: *log(n)* comparaciones para buscar *x* en el vector de nodos y otras *n* en la columna de *x* en la matriz de adyacencias para contar los arcos que llegan a *x*.
- *Implementación encadenada*: *n* comparaciones para buscar *x* en la lista de vértices, y otras *n* para recorrer la lista de arcos de *x* y contar aquellos que tienen a *x* como vértice final.

d.
- *Implementación matricial*: *n* comparaciones para buscar *x* en el vector de nodos, y otras n^2 en la matriz de adyacencias para contar los arcos que llegan a *x*.
- *Implementación encadenada*: *n* comparaciones para buscar *x* en la lista de vértices, y otras *m* para recorrer la lista de arcos de *x* y contar aquellos que tienen a *x* como vértice final.

Pregunta **4**

Correcta

Puntúa 1 sobre 1

En la columna de la izquierda se enumeran algunos algoritmos y estrategias que permiten resolver el problema del Corte Mínimo de un grafo G con n vértices y m arcos. Para cada uno, seleccione en la columna de la derecha la expresión de orden que corresponde al tiempo de ejecución para el peor caso en cada algoritmo. Una aclaración: el símbolo ^ debe entenderse como "elevar a la potencia indicada"... por ejemplo: *2^n* debe entenderse como 2^n (dos a la n).

Algoritmo de *Edmonds-Karp*	O(n * m^2) ✔
Algoritmo de *Goldberg-Tarjan*	O(n * m * log(n^2 / m)) ✔
Algoritmo de *Karger* (con $T = n^2 * ln(n)$ repeticiones)	O(n^2 * m * log(n)) ✔
Algoritmo de Fuerza Bruta	O(2^n) ✔

Pregunta **5**

Correcta

Puntúa 1 sobre 1

Sabemos que el ***algoritmo de Prim*** permite obtener un árbol de expansión mínimo para un grafo G *conexo*, ponderado, no dirigido y posiblemente denso. ¿Qué pasaría si el algoritmo (basado en el modelo de pseudocódigo que se planteó en el curso) se aplicase a un grafo *no conexo*?

Seleccione una:

a. El algoritmo no funcionaría, ya que nunca llegaría a tener todos los vértices de G en el AEM parcial. ✔

b. El algoritmo funcionaría y calcularía un árbol de expansión mínimo *separado* para cada componente conexa de G.

c. El algoritmo funcionaría, ya que agregaría arcos ficticios para unir los árboles de expansión mínimos de las componentes conexas de G.

d. El algoritmo no funcionaría, ya que retornaría un AEM exactamente igual a G.

Pregunta **6**

Correcta

Puntúa 1 sobre 1

Sabemos que el ***algoritmo de Kruskal*** permite obtener un árbol de expansión mínimo para un grafo G *posiblemente conexo*, ponderado, no dirigido y posiblemente denso. ¿Qué pasaría si el algoritmo (basado en el modelo de pseudocódigo que se planteó en el curso) se aplicase a un grafo *no conexo*?

Seleccione una:

- a. El algoritmo funcionaría y calcularía un árbol de expansión mínimo *separado* para cada componente conexa de G. ✔
- b. El algoritmo funcionaría, ya que agregaría arcos ficticios para unir los árboles de expansión mínimos de las componentes conexas de G.
- c. El algoritmo no funcionaría, ya que retornaría un AEM exactamente igual a G.
- d. El algoritmo no funcionaría, ya que nunca llegaría a tener todos los vértices de G en el AEM parcial.

Pregunta **7**

Correcta

Puntúa 1 sobre 1

Suponga que se propone la siguiente variante para el ***algoritmo de Kruskal***, para aplicar sobre un grafo *G* con *n* vértices y *m* arcos, conexo, ponderado, no dirigido y posiblemente denso:

- Ordene los m arcos de **mayor a menor** por pesos.
- Sea un grafo auxiliar T = G para ir armando el AEM parcial por descarte de arcos.
- Para k en [1..m] y (cantidad de arcos en T) != n - 1:
  - o Elimine el arco a_k de T.
- retorne T.

¿Qué se puede decir respecto de la variante propuesta?

Seleccione una:

- a. La variante funciona correctamente.
- b. La variante funciona siempre que el grafo sea efectivamente denso.
- c. Es imposible decidir si la variante propuesta funciona o no.
- d. La variante no funciona. ✔

Pregunta **8**

Correcta

Puntúa 1 sobre 1

¿Cuáles son las estrategias en las que se basa la implementación de las estructuras *Union-Find* presentada en el curso?

Seleccione una:

- a. Equilibrado de pesos y compresión de caminos. ✔
- b. Equilibrado de pesos y escisión.
- c. Equilibrado de alturas y división.
- d. Equilibrado de alturas y compresión de caminos.

Pregunta **9**

Correcta

Puntúa 1 sobre 1

¿Cuál de las siguientes sería una estrategia correcta para contar las componentes conexas de un grafo *G* no dirigido, con *n* vértices y *m* arcos?

Seleccione una:

- a. Crear una estructura Union-Find *T* con ***n*** componentes. Tomar uno a uno los *m* arcos *a(u, v)* de *G*, y controlar si *u* y *v* pertenecen ya a la misma clase en *T*. *Si no pertenecen a la misma clase*, hacer *union(u, v)* en *T*. Cuando haya finalizado el proceso para los *m* arcos, retornar la cantidad de grupos o clases que quedaron en *T*. ✔
- b. Crear una estructura Union-Find *T* con ***m*** componentes. Tomar uno a uno los *m* arcos *a(u, v)* de *G*, y controlar si *u* y *v* pertenecen ya a la misma clase en *T*. *Si no pertenecen a la misma clase*, hacer *union(u, v)* en *T*. Cuando haya finalizado el proceso para los *m* arcos, retornar la cantidad de grupos o clases que quedaron en *T*.
- c. Crear una estructura Union-Find *T* con ***m*** componentes. Tomar uno a uno los *m* arcos *a(u, v)* de *G*, y controlar si *u* y *v* pertenecen ya a la misma clase en *T*. Si pertenecen a la misma clase, hacer *union(u, v)* en *T*. Cuando haya finalizado el proceso para los *m* arcos, retornar la cantidad de grupos o clases que quedaron en *T*.
- d. Crear una estructura Union-Find *T* con ***n*** componentes. Tomar uno a uno los *m* arcos *a(u, v)* de *G*, y controlar si *u* y *v* pertenecen ya a la misma clase en *T*. *Si pertenecen a la misma clase*, hacer *union(u, v)* en *T*. Cuando haya finalizado el proceso para los *m* arcos, retornar la cantidad de grupos o clases que quedaron en *T*.

Pregunta **10**

Incorrecta

Puntúa 0 sobre 1

Sea un grafo $G = \{V, A\}$ donde V es el conjunto de n vértices y A es el conjunto de m arcos. Se define como el *grado de un vértice x* del grafo G a la cantidad de arcos incidentes en x (es decir, la cantidad de arcos que tienen a x como uno de sus extremos, sin importar si el arco es dirigido o no dirigido). Considere tanto la implementación matricial del grafo como la implementación por listas de adyacencia que se sugirió en el curso. ¿Cuál es en el peor caso el tiempo de ejecución en notación *Big O* de un método que calcule el *grado* de un vértice x cualquiera, en ambas implementaciones? (Observación: se supone que x es el objeto que se almacena en el vértice [y por lo tanto, en ambas implementaciones debe primero buscarlo en el vector de nodos o en la lista de vértices] y que el grafo puede ser dirigido o no dirigido, indistintamente a los efectos de la pregunta).

Seleccione una:

- a. Implementación matricial: $O(n^2)$ - Implementación encadenada: $O(n)$
- b. Implementación matricial: $O(n)$ - Implementación encadenada: $O(n)$
- c. Implementación matricial: $O(n^3)$ - Implementación encadenada: $O(n + m)$
- d. Implementación matricial: $O(n)$ - Implementación encadenada: $O(n + m)$ ✖

◄ Materiales Adicionales para la Semana 09

Ir a...

Ficha 10 [Algoritmos de Base Aleatoria] ►