Comenzado el	viernes, 6 de octubre de 2023, 12:00
Estado	Finalizado
Finalizado en	viernes, 6 de octubre de 2023, 12:14
Tiempo empleado	14 minutos 12 segundos
Calificación	**9,50** de 10,00 (**95**%)

Pregunta 1

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuál es el coste de añadir una arista en un grafo no dirigido con V vértices y A aristas, representado mediante una *matriz de adyacencia*?

Seleccione una:

- a. $O(V*A)$
- b. $O(1)$ ✔
- c. $O(V+A)$
- d. $O(V)$

Para añadir la arista $u-v$ basta con cambiar dos posiciones de la matriz ($G[u][v]$ y $G[v][u]$) a cierto.

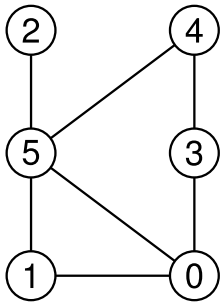
La respuesta correcta es: $O(1)$

Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

¿En qué orden se visitarían los vértices de este grafo si realizamos un recorrido en profundidad desde el vértice 1? Escribe los identificadores de los vértices separados por espacios en el orden en que son visitados. Supón que los vértices en las listas de adyacentes están ordenados de menor a mayor.



Respuesta: 1 0 3 4 5 2 ✔

Los vértices se recorren en este orden: 1 0 3 4 5 2.

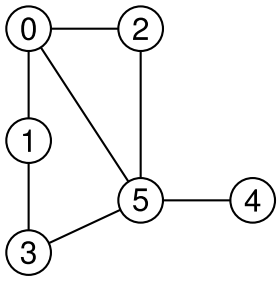
La respuesta correcta es: 1 0 3 4 5 2

Pregunta 3

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuáles son posibles recorridos en anchura de este grafo comenzando el recorrido en el vértice 0? No sabemos en qué orden aparecen los adyacentes en cada lista de adyacentes.



Seleccione una o más de una:

- a. 0 5 2 1 3 4 ✔ Cierto.
- b. 0 2 5 1 4 3 ✔ Cierto.
- c. 0 1 5 2 4 3
- d. 0 1 2 3 5 4

a. Cierto.
b. Cierto.
c. Falso. Si el 1 se visita antes que el 5 como adyacentes del vértice 0, entonces el vértice 1 entrará antes en la cola y por tanto el vértice 4 no puede visitarse antes que el 3.
d. Falso. El vértice 3 (a distancia dos del origen) no puede visitarse antes que el 5 (a distancia uno).

Las respuestas correctas son: 0 5 2 1 3 4, 0 2 5 1 4 3

Pregunta 4

Correcta

Se puntúa 1,00 sobre 1,00

¿Qué complejidad tendría un algoritmo para calcular el grado de cada uno de los vértices de un grafo no dirigido representado mediante *listas de adyacentes* si el grafo tiene V vértices y A aristas?

Seleccione una:

- a. $O\left(V^2\right)$
- b. $O(V)$ ✔
- c. $O(V*A)$
- d. $O(V+A)$

Calcular el grado de un vértice tiene coste en $O(1)$ (los adyacentes están ya almacenados en una lista, y solamente necesitamos la longitud de la lista, no recorrerla). Y esta operación se repite para cada vértice. El coste total está en $O(V)$.

La respuesta correcta es: $O(V)$

Pregunta 5

Correcta

Se puntúa 1,00 sobre 1,00

En un grafo no dirigido, para encontrar el camino más corto (con menos aristas) entre dos vértices se utiliza

Seleccione una:

- a. ni búsqueda en profundidad ni en anchura.
- b. búsqueda en profundidad.
- c. búsqueda en anchura. ✔
- d. tanto búsqueda en profundidad como en anchura.

La búsqueda en anchura, al ir recorriendo los vértices por distancias crecientes desde el origen, garantiza que encuentra el camino con menos aristas.

La respuesta correcta es: búsqueda en anchura.

Pregunta 6

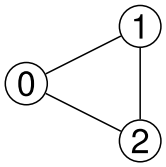
Parcialmente correcta

Se puntúa 0,50 sobre 1,00

Si en una búsqueda en profundidad sobre un grafo no dirigido *G* se desmarcase el vértice *v* al salir de la llamada recursiva *dfs*(*G*, *v*), ¿cuáles de las siguientes afirmaciones serían ciertas?

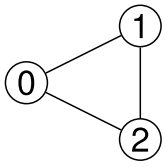
Seleccione una o más de una:

- a. La búsqueda podría no terminar. ✖ Falso. El número de llamadas recursivas anidadas no puede superar el número de vértices, porque estos siguen marcados hasta que acaban sus llamadas recursivas. Cada llamada recursiva hace a lo sumo tantas llamadas como el número de aristas (o de vértices), por lo que el número de llamadas totales es finito y el algoritmo termina.
- b. Un mismo vértice podría visitarse más de una vez. ✔ Cierto. Por ejemplo, en el grafo siguiente, el algoritmo de búsqueda desde 0 visitaría en este orden 0 1 2 2 1, repitiendo los vértices 2 y 1 que han sido desmarcados en el momento de volverlos a visitar.



a. Falso. El número de llamadas recursivas anidadas no puede superar el número de vértices, porque estos siguen marcados hasta que acaban sus llamadas recursivas. Cada llamada recursiva hace a lo sumo tantas llamadas como el número de aristas (o de vértices), por lo que el número de llamadas totales es finito y el algoritmo termina.

b. Cierto. Por ejemplo, en el grafo siguiente, el algoritmo de búsqueda desde 0 visitaría en este orden 0 1 2 2 1, repitiendo los vértices 2 y 1 que han sido desmarcados en el momento de volverlos a visitar.



La respuesta correcta es: Un mismo vértice podría visitarse más de una vez.

Pregunta 7

Correcta

Se puntúa 1,00 sobre 1,00

Si el grafo está representado mediante una *matriz de adyacencia*, ¿cuál es la complejidad del siguiente algoritmo que calcula el número de aristas del grafo si este tiene V vértices y A aristas?

```
Grafo grafo(V);
int aristas = 0;
for (int v = 0; v < V; ++v)
   aristas += grafo.ady(v).size();
cout << aristas/2 << '\n';
```

Seleccione una:

a. $O(V+A)$

b. $O(V)$

c. $O(V*A)$

d. $O\left(V^2\right)$ ✔

La operación para calcular los adyacentes a un vértice tiene coste en $O(V)$ (aunque después de la lista solamente queramos saber su longitud). Y esta operación se repite para cada vértice. El coste total está en $O\left(V^2\right)$.

La respuesta correcta es: $O\left(V^2\right)$

Pregunta 8

Correcta

Se puntúa 1,00 sobre 1,00

El coste de la operación que averigua si dos vértices son adyacentes en un grafo no dirigido con V vértices y A aristas, cuando se utiliza la representación mediante *listas de adyacentes* es del orden de:

Seleccione una:

a. $O(V)$ ✔

b. $O(\log V)$

c. $O(V \log V)$

d. $O\left(V^2\right)$

Para saber si los vértices u y v son adyacentes hay que recorrer la lista de adyacentes al vértice u buscando a v o al revés. La longitud de esas listas está acotada por el número de vértices (si no hay aristas repetidas).

La respuesta correcta es: $O(V)$

Pregunta 9

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuántos árboles tiene un bosque (grafo cuyas componentes conexas son todas árboles libres) de 46 vértices y 32 aristas?

Seleccione una:

- a. 14 ✔
- b. 13
- c. 45
- d. No se puede saber.

Al saber que las componentes conexas son árboles libres, es decir, no tienen ciclos, si vamos añadiendo las aristas una a una, cada arista nueva reduce en 1 el número de árboles (conecta dos árboles). Por tanto, con 46 vértices y 32 aristas tenemos (46 − 32 =) 14 árboles libres en el bosque.

La respuesta correcta es: 14

Pregunta 10

Correcta

Se puntúa 1,00 sobre 1,00

Queremos representar en un grafo todos los habitantes de un pueblo de tal forma que haya una arista entre dos personas si son parientes de cualquier grado. Además, cuando nos pregunten si dos personas son parientes queremos responder muy rápidamente. ¿Cuál sería la representación más conveniente para el grafo?

Seleccione una:

- a. Listas de adyacencia.
- b. Matriz de adyacencia. ✔

Si queremos responder muy rápidamente a la pregunta de si dos personas son parientes, es decir, si existe arista entre dos vértices, necesitamos una matriz de adyacencia donde esta operación tiene coste constante en lugar de lineal sobre el número de aristas.

La respuesta correcta es: Matriz de adyacencia.