



## Complejidad Parcial EA 2024 1

Complejidad Algorítmica (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu

### Pregunta 1

1.5 de 1.5 puntos

¿Cuánto espacio es necesario para almacenar una matriz de adyacencia, en función a  $n$ , el número de vértices y  $m$ , el número de aristas?

Elige la respuesta correcta.

Respuesta seleccionada: ☒ b.  $O(n^2)$

Respuestas:

a.  $O(m+n)$

b.  $O(n^2)$

c.  $O(m)$

d.  $O(n)$

### Pregunta 2

0 de 1.5 puntos

Señale el/los algoritmo(s) que invierte(n) el grafo para encontrar su solución.

Respuestas seleccionadas: ☒ d. SCC (Componentes Fuertemente Conexas)

Respuestas:

a. DFS

b. BFS

c. Orden Topológico

d. SCC (Componentes Fuertemente Conexas)

### Pregunta 3

0.75 de 1.5 puntos

Cuando se ejecuta para todas las nodos, es de tiempo  $O(n+m)$

Respuestas seleccionadas: ☒ a. Orden Topológico

Respuestas:

b. BFS

a. Orden Topológico

b. BFS

c. SCC (Componentes Fuertemente Conexas)

d. DFS

### Pregunta 4

1.5 de 1.5 puntos

¿Cuál es el propósito principal de utilizar el ordenamiento topológico en la teoría de grafos?

Respuesta seleccionada: ☒ d. Resolver problemas de planificación y dependencias.

Respuestas:

a. Calcular la distancia más corta entre dos nodos.

b. Encontrar ciclos en grafos dirigidos.

c. Determinar la conectividad de un grafo.

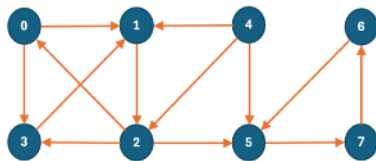
d. Resolver problemas de planificación y dependencias.

### Pregunta 5

0 de 6 puntos

El siguiente grafo representa a un grupo de 8 personas que pertenecen a una red social.

Cada persona es representada por un Nodo o Vértice en el Grafo. La etiqueta del nodo es el identificador de la persona en la red social. El grafo nos brinda información acerca de cómo una persona es seguidora (follower) de otra persona en la red social. Así, por ejemplo, podemos interpretar que la Persona\_0 sigue a la Persona\_1 y la Persona\_7 es seguida por la Persona\_6.



Se solicita desarrollar un código en lenguaje Python que permita:

- (1 punto) Indicar el algoritmo o técnica utilizado para resolver el requerimiento.
- (5 puntos) Identificar los grupos de personas que estén estrechamente vinculados debido a sus múltiples relaciones, gustos, lugares comunes y preferencias en la red social, de tal forma que luego se pueda usar este algoritmo en un sistema que pueda gestionar recomendaciones para quienes conforman estos componentes.
  - La salida o resultado debe considerar:
    - Visualizar el grafo dirigido original.
    - Mostrar el resultado de la variable conteniendo los grupos de personas.
    - Visualizar el grafo dirigido resultante.

Añade su archivo respuesta en formato ".ipynb" o ".py". No olvides adjuntar el archivo de tipo imagen correspondiente a la visualización (de ser el caso).

### Pregunta 6

0 de 6 puntos

En un centro de estudios se ha detectado un nuevo brote del Covid-20 y para minimizar el contagio, la mejor estrategia es separar a los sospechosos del resto de estudiantes.

Es sabido que en el centro de estudios existen muchos grupos de estudiantes, así, en un mismo grupo, los estudiantes se relacionan con mucha frecuencia, pero además, son libres de relacionarse o unirse a otros grupos.

Para prevenir un posible contagio masivo, el centro de estudios ha elaborado una lista de todos los grupos de estudiantes para aplicar la siguiente regla:

**UNA VEZ QUE UN MIEMBRO DE UN GRUPO ES SOSPECHOSO, TODOS LOS MIEMBROS DEL GRUPO TAMBIÉN LO SON.**

El problema es que no es tan fácil identificar a todos los sospechosos cuando un estudiante es catalogado como tal, por tanto, su trabajo será escribir un programa Python el cual encuentre a todos los sospechosos.

Como datos de entrada, se le proporciona un archivo de texto:

[suspicious\\_students.txt](#)

Este texto puede contener varios casos de grupos de estudiantes a evaluar.

Ejemplo de entrada:

```
100 4
2 1 2
5 10 13 11 12 14
2 0 1
2 99 2
200 2
1 5
5 1 2 3 4 5
1 0
0 0
```

Donde:

- Se considera una línea de inicio de un nuevo caso a evaluar, cuando sólo existen dos números enteros separados por un espacio (las líneas en negrita).
  - El primer número representa la cantidad de estudiantes y lo identificamos por la variable  $n$  (donde  $n$  puede tomar valores de 0 a 30000).
  - El segundo número es la cantidad de grupos de estudiantes. Este número lo representaremos por la letra  $m$  (donde  $m$  puede tomar valores de 0 a 500).
- A continuación de la línea de inicio de un nuevo caso le siguen  $n$  líneas. Cada línea identifica a una lista de miembros por grupo.
  - El primer número de cada lista de miembros lo llamaremos  $k$ , y representa el número de miembros en el grupo.
  - Los índices de estudiantes  $k+1$  y separados por un espacio, representan el identificador de los estudiantes. Inicialmente, los estudiantes son identificados por un único valor entero que está entre 0 y  $n-1$ . El estudiante identificado con el número 0 es reconocido como un sospechoso en cada caso que se evalúe.
- La última línea o la condición de fin de lectura del archivo debe validar que  $n = m = 0$ . A partir de esta condición de salida, se deberá calcular y obtener como resultado:
  - El número de casos evaluados.
  - El total de sospechosos encontrados por cada caso evaluado.

Ejemplo de salida:

```
Número de casos evaluados: 3
#Sospechosos - Caso #1: 4
#Sospechosos - Caso #2: 1
#Sospechosos - Caso #3: 1
```

Se solicita:

- (1 punto) Desarrollar un código en Python que lee el archivo de texto y produzca la salida deseada.
- (1 punto) Indicar que algoritmo tomó de base para solucionar este ejercicio y por qué.
- Se tendrá en cuenta para la calificación el orden y la explicación (documentación) de su solución.

Añade su archivo respuesta en formato ".ipynb" o ".py" (si hace uso de alguna visualización, no olvide adjuntar también el archivo de la imagen y el txt).