UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN

Taller de programación II - IN1071C

Algoritmos en Grafos

Diego Maldonado

Facultad de Ingeniería
Departamento de Ingeniería Informática

19 de octubre de 2022

Contenido

Definición (Grafo)

Un grafo es un par ordenado $G = (V, E)$ donde:

V: conjunto no vacio y finito de elementos llamados vértices.

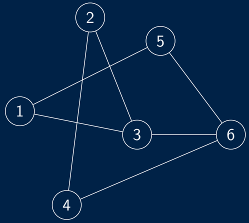
E: conjunto de pares no ordenados de V llamados aristas. Dicho de otra forma ($E \subseteq \mathcal{P}(V)$ y $\forall e \in E, |e| = 2$)

Definición

Notación

- $V(G)$ conjunto de vértices
- $E(G)$ conjunto de aristas
- $\{u, v\} = uv = vu = e \in E(G)$ arista con $u, v \in V(G)$
- $|V(G)|$ número de vértices
- $|E(G)|$ número de aristas
- $\{u, v\} \in E \Leftrightarrow$ u es adyacente a v

Representación



$$V = \{1, 2, 3, 4, 5, 6\}$$
$$E = \{\{1, 3\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{3, 6\}, \{4, 6\}, \{5, 6\}\}$$

Definición

Dado un grafo $G = (V, E)$ Definimos

- la vecindad de v como:
  $N_G(v) := \{u \in V : u \text{ es vértice adyacente a } v \text{ en } G\}$
- el grado de $u \in V$ como $d_G(u) := |N_G(u)|$

Definición (Digrafo)

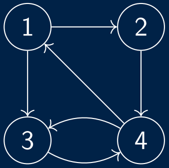
Un grafo dirigido o digrafo es un par ordenado $G = (V, E)$ donde:

- V: conjunto no vacío y finito de elementos llamados vértices.
- E: conjunto de pares ordenados de V llamados aristas dirigidas. Dicho de otra forma $E \subseteq V \times V$

Representaremos las aristas dirigidas como flechas

Ejemplo

$V = \{1, 2, 3, 4\}$
$E = \{(1, 2), (1, 3), (2, 4), (3, 4), (4, 3), (4, 1)\}$



Definición

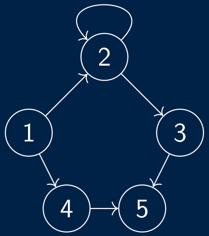
Dado un digrafo $G = (V, E)$ Definimos

- u apunta o incide en v si $(u, v) \in E$
- $e \in E$ es un bucle si $e = (u, u)$
- $N_G^+(v) := \{u \in V : u \text{ apunta } v\}$
- $N_G^-(v) := \{u \in V : u \text{ es apuntado por } v\}$

Ejercicio

Para el siguiente grafo determine

- $N^+(1)$ y $N^-(1)$
- Si hay algún bucle
- El vértice que apunta a más vértices



Definición

Dado un digrafo $G = (V = \{v_1, ..., v_n\}, E)$ con n vértices, definimos su matriz de adyacencia $A(G) \in \mathcal{M}_n(\{0, 1\})$ como la matriz de componentes

$$a_{i,j} = \begin{cases} 1, & \text{si } (v_i, v_j) \in E \\ 0, & \text{en otro caso.} \end{cases}$$

Si G es un grafo no dirigido, entonces consideraremos el digrafo $\vec{G}$ donde que cada arista de G es reemplazada por dos aristas dirigidas apuntando en direcciones opuestas.

Ejercicio

Considere la siguiente matriz

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Construya una representación gráfica de esta matriz.

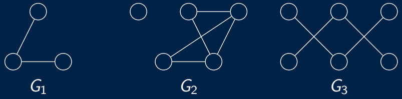
Definición

Dado un grafo $G = (V, E)$ definimos

- $u, v \in V$ están conectados si existe un camino de u a v.
- G se dice conexo $|V| = 1$ o $\forall u, v \in V$ u y v están conectados.
- Si G no es conexo se dice disconexo.
- Una componente conexa de G es un sub-grafo maximal en la propiedad de conexidad, es decir, no existe otro sub-grafo conexo que contenga estrictamente a alguna componente conexa.

Ejercicio

- ¿Cuáles de los siguientes grafos son conexos?
- ¿Cuántas componente conexas tiene estos grados y cuáles son?



Existen dos grandes formas de recorrer un grafo conexo, la búsqueda en profundidad y la búsqueda en anchura. Cuando el grafo no es conexos, estas formas de recorrer el grafo no logran visitar todos los vecinos, pues se basan en recorrer el grafo pasando de un vértice a algún vecino. La búsqueda en profundidad se basa en partir de un vértice tratar de avanzar lo más posible sin repetir vértices. Su algoritmo es el siguiente:

Búsqueda en profundidad (DFS, Depth First Search)

Entrada: $G(V, E)$, grafo no dirigido, $v_0 \in V$

Salida: visitado, lista de vértices alcanzables desde v_0

booleano visitado $[1 \dots n]$
Para $u \in V$ **hacer**
  visitado $[u] \leftarrow$ *Falso*
$DFS(v_0)$
; ▷ Búsqueda en profundidad (Depth First Search)
Sub-rutina $DFS(u)$:
  visitado $[u] \leftarrow$ *Verdadero*
  Para cada $v \in N(u)$ **hacer**
    Si $visitado[v] = Falso$ **entonces**
      $DFS(v)$

Existen dos grandes formas de recorrer un grafo conexo, la búsqueda en profundidad y la búsqueda en anchura. Cuando el grafo no es conexos, estas formas de recorrer el grafo no logran visitar todos los vecinos, pues se basan en recorrer el grafo pasando de un vértice a algún vecino. La búsqueda en profundidad se basa en partir de un vértice tratar de avanzar lo más posible sin repetir vértices. Su algoritmo es el siguiente:

Búsqueda en profundidad (DFS, Depth First Search)

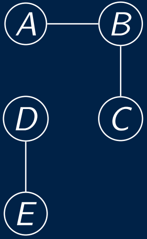
Entrada: $G(V, E)$, grafo no dirigido, $v_0 \in V$

Salida: visitado, lista de vértices alcanzables desde v_0 , lista de predecesores para llegar a v_0;

booleano visitado $[1 \ldots n]$
Para $u \in V$ **hacer**
  visitado $[u] \leftarrow$ *Falso* ; predecesor$[u] \leftarrow \perp$
predecesor $[v_0] \leftarrow v_0$; $DFS(v_0)$
; ▷ Búsqueda en profundidad (Depth First Search)
Sub-rutina $DFS(u)$:
  visitado $[u] \leftarrow$ *Verdadero*
  Para cada $v \in N(u)$ **hacer**
    Si *visitado*$[v]$ = *Falso* **entonces**
      predecesor $[v] \leftarrow u$; $DFS(v)$

UVA 00459 - Graph Connectivity (Enunciado)

Consideremos un grafo G formado por un gran número de vértices conectados por aristas. Se dice que G es conexo si se puede encontrar un camino en 0 o más pasos entre cualquier par de nodos en G. Por ejemplo, el siguiente grafo no es conexo porque no hay ningún camino de A a C.



Sin embargo, este grafo contiene una serie de subgrafos conexos, uno para cada uno de los siguientes conjuntos de vértices: $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, $\{A, B\}$, $\{B, D\}$, $\{C, E\}$, $\{A, B, D\}$

Un subgrafo conexo es maximal si no hay nodos y aristas en el grafo original que puedan añadirse al subgrafo y seguir dejándolo conectado. Hay dos subgrafos conexos maximales arriba, uno asociado con los nodos $\{A, B, D\}$ y el otro con los nodos $\{C, E\}$.

Escriba un programa para determinar el número de subgrafos conexos maximales de un grafo dado.

UVA 00459 - Graph Connectivity (Entrada-Salida)

Entrada

La entrada comienza con un único número entero positivo en una línea por sí mismo indicando el número de los casos siguientes, cada uno de ellos como se describe a continuación. Esta línea va seguida de una línea en blanco, y también hay una línea en blanco entre dos entradas consecutivas.

La primera línea de cada conjunto de entradas contiene un único carácter alfabético en mayúsculas. Este carácter representa el nombre del nodo más grande del gráfico. Cada línea sucesiva contiene un par de caracteres alfabéticos en mayúscula que representan una arista del grafo.

La sección de entrada de ejemplo contiene un posible conjunto de entradas para el grafo representado arriba. La entrada se termina con una línea en blanco.

Salida

Para cada caso de prueba, escriba en la salida el número de subgrafos conexos maximales. Las salidas de dos casos consecutivos estarán separadas por una línea en blanco.

UVA 00459 - Graph Connectivity (Entrada-Salida)

Entrada de Prueba

```
1

E
AB
CE
DB
EC
```

Salida de Prueba

```
2
```

UVA 00469 - Los Humedales de Florida (Enunciado)

Una empresa de construcción es propietaria de un gran terreno en el estado de Florida. Recientemente, la empresa decidió desarrollar esta propiedad. Sin embargo, al inspeccionar la propiedad, se descubrió que el terreno, en varios lugares, contenía masas de agua. Esto sorprendió a los propietarios de la empresa, ya que eran de fuera del estado y no estaban familiarizados con los humedales de Florida. La situación era muy grave y los propietarios no sabían que esas masas de agua pueden convertirse en hermosos lagos que aumentarán el valor de los terrenos que los rodean, estuvieron a punto de abandonar el proyecto de construcción. Afortunadamente, un inteligente graduado de la FIU que trabajaba para la empresa llamó la atención de los propietarios sobre este hecho y, en consecuencia, el proyecto de construcción se llevó a cabo.
Los ingenieros dividieron la obra en una cuadrícula uniforme, de modo que cada celda de la cuadrícula contenía agua o tierra.(Cómo lo hicieron, por supuesto, es una incógnita.) Ahora, la pregunta que deben responder los ingenieros es la siguiente: "Dado el número de fila y columna de una celda de la cuadrícula que contiene agua, cuál es el área del lago que contiene esa celda". (un área se mide por el número de celdas de la cuadrícula que contiene. Las celdas diagonales se consideran adyacentes).
Debe escribir un programa para responder a esta pregunta.

UVA 00469 - Los Humedales de Florida (Entrada-Salida)

Entrada

La entrada comienza con un único número entero positivo en una sola línea que indica el número de los casos siguientes, cada uno de ellos como se describe a continuación. Esta línea va seguida de una línea en blanco, y también hay una línea en blanco entre dos entradas consecutivas.
La entrada consiste en $0 < n \leq 99$ líneas, cada una de las cuales contiene $0 < m \leq 99$ caracteres de secuencia de 'L's y 'W's seguida de $k > 0$ líneas cada una de las cuales contiene un par de enteros i y j. Las primeras n líneas representan la cuadrícula $n \times m$ que cubre el terreno donde una 'W'/'L' en el carácter c-ésimo de la línea r-ésima indica agua/tierra dentro de la celda de la fila r y la columna c de la cuadrícula. Los pares de enteros de las últimas k líneas representan los números de fila y columna de alguna celda de la cuadrícula que contiene agua.

Salida

Para cada caso de prueba, la salida debe seguir la siguiente descripción. Las salidas de dos casos consecutivos irán separadas por una línea en blanco. La salida para cada par de enteros, i y j, en las últimas k líneas de entrada, consiste en un entero, en una línea separada, que indica el área del lago que contiene la celda de la cuadrícula, en la fila i y la columna j de la cuadrícula.

UVA 00469 - Los Humedales de Florida (Entrada-Salida)

Entrada de Prueba

```
1

LLLLLLLLL
LLWWLLWLL
LWWLLLLLL
LWWWLWWLL
LLLWWWLLL
LLLLLLLLL
LLLWWLLWL
LLWLWLLLL
LLLLLLLLL
3 2
7 5
```

Salida de Prueba

```
12
4
```