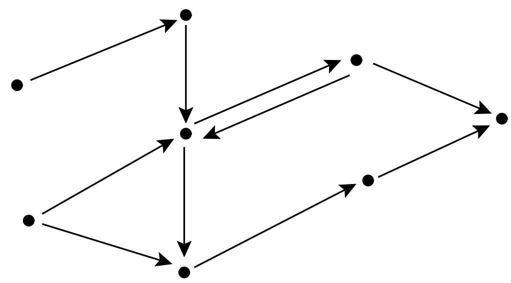
Investigación Operativa - Optimización Combinatoria

Práctica 3

1. Dado el grafo dirigido



i) Hallar su matriz de incidencia vértice-rama.

ii) Determinar si el grafo es fuertemente conexo.

iii) Indicar si existe algún ciclo dirigido.

2. Sea G un grafo dirigido cuya matriz de incidencia vértice-rama sea

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

i) Sin graficar G, determine si G contiene algún ciclo dirigido.

ii) Grafique G y verifique que su respuesta a i) fue correcta.

3. Sea $G = (V, E)$ un grafo no dirigido.

i) Probar que G es un árbol si y sólo si para todo $u, v \in V$, $u \neq v$, existe un único camino que une u y v.

ii) Probar que si G es un árbol entonces cuando le quitamos una rama deja de ser conexo.

iii) Probar que si G es conexo entonces $\#V \leq \#E + 1$. ¿Cuándo vale la igualdad?

iv) Probar que la cantidad de componentes conexas de G es mayor o igual que $\#V - \#E$ y que vale la igualdad si y sólo si G es acíclico.

4. Sea $G = (V, E)$ el grafo no dirigido donde $V = \{2, 3, 4, 5, 6, \ldots, 11\}$ y E está definido por

$$(u, v) \in E \quad \text{si y sólo si} \quad u \text{ y } v \text{ son coprimos}$$

i) Hallar su tabla de adyacencia.

ii) Aplicando el algoritmo breadth-first search, hallar un camino de $s = 2$ a t para cada $t \in V$ tal que exista un camino de s a t.

iii) A partir del resultado del algoritmo, determine si G es conexo.

5. Sea $G = (V, E)$ el grafo no dirigido donde $V = \{2, 3, 4, 5, 6, \ldots, 16\}$ y E está definido por

$$(u, v) \in E \quad \text{si y sólo si} \quad u \text{ es divisible por } v \text{ o } v \text{ es divisible por } u$$

i) Hallar su tabla de adyacencia.

ii) Aplicando el algoritmo depth-first search, hallar un camino de $s = 4$ a t para cada $t \in V$ tal que exista un camino de s a t.

iii) A partir del resultado del algoritmo, determine si G es conexo.

6. ¿Cómo usaría el algoritmo search para determinar si un grafo no dirigido es conexo?

7. ¿Cómo usaría el algoritmo search para determinar si un grafo dirigido

i) es conexo?

ii) es fuertemente conexo?

8. Sea $G = (V, E)$ el grafo no dirigido donde $V = \{2, 3, 4, 5, 6, \ldots, 26\}$ y E está definido por

$$(u, v) \in E \quad \text{si y sólo si} \quad u \text{ y } v \text{ tienen un divisor común}$$

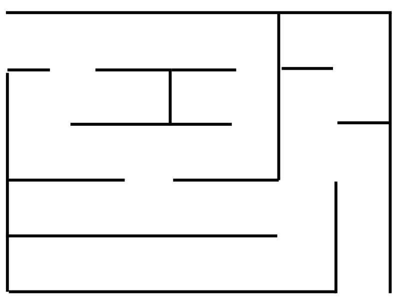
Aplicar el algoritmo search para hallar la componente conexa que contiene a $u = 2$.

9. Modificar convenientemente breadth-first search para obtener un algoritmo de complejidad polinomial que determine cuántas y cuáles son las componentes conexas de un grafo no dirigido dado.

Nota: dado que un grafo no dirigido G es acíclico si y sólo si tiene $\#V - \#E$ componentes conexas (ver ejercicio 3. iv)), observar que este algoritmo determina si un grafo no dirigido es acíclico en tiempo polinomial.

10. Demuestre la validez del algoritmo search.

11. Considere en el siguiente laberinto



el problema de hallar la salida. Haga un planteo utilizando un grafo representativo del laberinto. ¿Qué algoritmo usaría para solucionar el problema?

12. Sea $G = (V, E)$ un grafo no dirigido, donde $V = \{1, 2, \ldots, m\}$. Considere el siguiente algoritmo

1. Poner $u = 1$ y $r_{ij} = \infty$ para todo i, j entre 1 y m.

2. Para cada $v \in V$ tal que $v > u$, hallar, si existe, un camino $\mathcal{C}_{uv}$ de u a v utilizando breadth-first search. Para los v tales que no exista, actualizar r_{uv} poniendo $r_{uv} = 0$.

3. Para cada $v \in V$ tal que $v > u$ y $r_{uv} \neq 0$ determinar, utilizando breadth-first search, si existe un camino de u a v en el grafo que resulta de eliminar de G todas las ramas de $\mathcal{C}_{uv}$, todos los vértices de $\mathcal{C}_{uv}$ distintos de u y de v y todas las ramas que inciden en ellos. Para los v tales que exista, actualizar r_{uv} poniendo $r_{uv} = 1$.

4. Poner $u = u + 1$. Si $u \leq m$ ir a 2. Si no STOP.

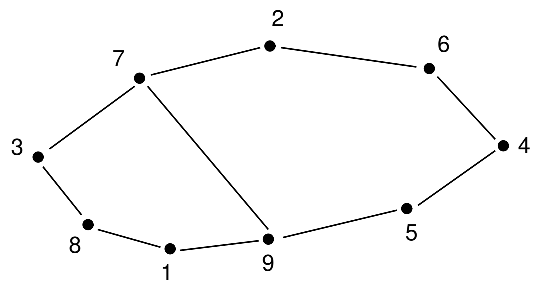
i) Probar que G es acíclico si y sólo si al finalizar el algoritmo $r_{ij} \neq 1$ para todo i, j.

ii) Probar que G es un árbol si y sólo si al finalizar el algoritmo $r_{ij} = \infty$ para todo i, j .

iii) Calcular la complejidad de este algoritmo.

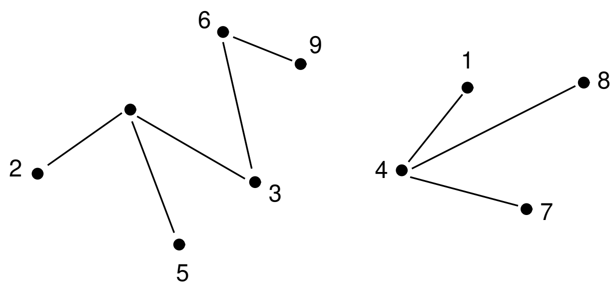
iv) Supongamos que G no es acíclico. ¿Es cierto que dados i y j $(i < j)$ entonces existe un ciclo de G que pasa por i y por j si y sólo si al finalizar el algoritmo $r_{ij} = 1$?

v) Aplicar el algoritmo al grafo



para comprobar que no es acíclico.

vi) Aplicar el algoritmo al grafo



para comprobar que es acíclico pero no es un árbol.

vii) ¿Siguen valiendo i) y ii) si en los pasos 2. y 3. del algoritmo se usa depth-first en lugar de breadth-first?

13. Sea $G = (V, E)$ un grafo no dirigido. Considere el siguiente algoritmo

1. Poner $r_e = \infty$ para todo $e \in E$.

2. Para cada $e = (u, v) \in E$ determinar, utilizando el algoritmo search, si existe un camino de u a v en el grafo que resulta de eliminar de G la rama e. Para los e tales que exista, actualizar r_e poniendo $r_e = 1$.

i) Probar que G es acíclico si y sólo si al finalizar el algoritmo $r_e = \infty$ para todo $e \in E$.

ii) Calcular la complejidad de este algoritmo.

iii) Supongamos que G no es acíclico. ¿Es cierto que dada una rama e entonces existe un ciclo de G que contiene a e si y sólo si al finalizar el algoritmo $r_e = 1$?

iv) Aplicar el algoritmo al grafo del ejercicio 12. v) para comprobar que no es acíclico.

v) Aplicar el algoritmo al grafo del ejercicio 12. vi) para comprobar que es acíclico.

14. Los algoritmos de los ejercicios 9., 13. y 14. determinan si un grafo no dirigido G es acíclico. ¿Cuál de ellos es el más eficiente?