Guía de Ejercicios – Parcial #2

Carlos Alberto Pérez

October 28, 2012

1 Caminos, ciclos y alcanzabilidad

1.1 Repaso e Investigación Teórica

1. Defina los siguientes conceptos:
   (a) Camino (dirigido y no dirigido).
   (b) Ciclo (dirigido y no dirigido).
   (c) Alcanzabilidad. *(Dependiendo del tipo de grafo, el concepto puede tener una variante.)*
   (d) Grafo conexo.
   (e) Componente conexa. *(El mismo concepto debe aplicarse a **ambos** tipos de grafos.)*
   (f) Componente fuertemente conexa. *(Este concepto se aplica sólo a grafos dirigidos.)*
2. Investigue los siguientes conceptos:
   (a) Puntos de artículación.
   (b) Puentes.
   (c) Componentes biconexas.

1.2 Ejercicios

1. Determine, considerando que el grafo a trabajar no puede tener lados múltiples:
   (a) La cantidad máxima de caminos simples que puede tener un grafo de orden n. *(Puede suponer que el grafo es completo.)*
   (b) La cantidad máxima de ciclos simples que puede tener un grafo de orden n.
   (c) La cantidad máxima de lados que puede tener un grafo dirigido para que todos los caminos resultantes sean simples.
2. Considere un ciclo $c =< n_1, e_1, n_2, e_2, n_3, \ldots, n_i, e_i, n_1 >$. En clase se habló de la operación de rotación, el cuál lo que hace es construir un nuevo cíclo, a partir de otro cíclo ya existente, tal que el nuevo cíclo tenga un nuevo nodo inicial (y por ende el mismo nuevo nodo como final) y que cruce los mismos lados. Existen dos rotaciones: la rotación a la derecha de c (llámemosla $\overrightarrow{c}$) que está definida como: $\overrightarrow{c} =< n_2, e_2, n_3, \ldots, n_i, e_i, n_1, e_1, n_2 >$, y la rotación a la izquierda de c cuya definición es: $\overleftarrow{c} =< n_i, e_i, n_1, e_1, n_2, e_2, n_3, \ldots, n_i >$.
   (a) ¿Cuántas rotaciones se puede realizar a un ciclo simple? *(Un ciclo simple es aquél en que no se repita ningún nodo exceptuando el inicial/final).* ¿Qué ocurre con las rotaciones si el ciclo no es simple?
   (b) Considere la operación $\overrightarrow{c^n}$ que corresponde a realizar n rotaciones a la derecha al ciclo c. Si la longitud del ciclo c es n, ¿es cierto que $\overrightarrow{c^i} = \overrightarrow{c^{i+n}}$?
   (c) Defina la relación $\equiv$ entre cíclos, tal que $c_1 \equiv c_2$ si se puede obtener c_2 rotando c_1. ¿Es $\equiv$ una clase de equivalencia? Justifique su respuesta.
   (d) Considere un ciclo que no es simple (es decir, aquél en que un nodo se repite, sin contar la repetición inherente en el nodo inicial). Argumente que, si un ciclo no es simple, es porque existe un ciclo simple contenido en el ciclo no simple que tiene como inicial al nodo repetido.

(e) Proponga una relación entre las clases de equivalencia de $\equiv$, llámemosla $\hookleftarrow$, tal que $c_1 \hookleftarrow c_2$ si se puede usar un elemento de la clase c_2 dentro de un elemento de la clase c_1 para construir un cíclo no simple. ¿Qué debe cumplir $\hookleftarrow$ para poder decir que un grafo no contenga ciclos no simples y no triviales? *(Un ciclo no simple trivial es aquél que se obtiene de repetir un ciclo simple más de una vez o es obtenido de recorrer un camino no dirigido en sentido inverso, por ejemplo, si c es un ciclo simple, entonces $c +\!\!+\, c$ es un ciclo no simple trivial.)*

3. Un camino (ciclo) se denomina Euleriano si el camino cruza por todos los lados exactamente una vez. En base a este concepto responda las siguientes preguntas: *(considere que el grafo es no dirigido)*

   (a) De un ejemplo de un camino (ciclo) euleriano que no sea simple.

   (b) ¿Existirá un camino (ciclo) euleriano si hay más de una componente conexa? De ser cierto, ¿cómo deben estar estructuradas dichas componentes?

   (c) ¿Existirá un ciclo euleriano si algún nodo tiene grado impar? ¿Puede postular una condición similar para un camino euleriano?

   (d) Argumente que de existir un ciclo euleriano, tal ciclo es simple si el grado de todos los nodos es 2.

   (e) Considere la definición de la relación $\hookleftarrow$ (definida por ud. en la pregunta 2e). ¿Qué condición debe cumplir c_1 y c_2 tal que, si $c_1 \hookleftarrow c_2$, entonces el ciclo representado resultante sea euleriano?

   (f) Use el resultado obtenido en la pregunta 3e para proponer un algoritmo que determine si un grafo tiene un ciclo euleriano. ¿Qué orden tiene su algoritmo?

   (g) Investigue acerca del algoritmo de Hierholzer [1]. ¿Cómo se compara su algoritmo contra el algoritmo de Hierholzer?

2 Recorridos en Grafos

2.1 Repaso e Investigación Teórica

1. Enuncie las tres fases que debe cumplir un algoritmo de recorrido en grafos.

2. ¿Cuáles son los dos posibles problemas que podemos trabajar en recorridos? Enuncie formas de trabajar un problema en función de otro, si existen.

3. Enuncie el esquema general de recorrido. ¿Por qué este esquema no se puede implementar directamente en una computadora?

4. Defina el concepto de estrategia de recorrido y argumente porque es necesario definir una estrategia.

5. Enuncie el algoritmo de Búsqueda en Profundidad (DFS, del inglés: *depth-first search*). Para este enunciado, solamente indique como realizar el recorrido, mas no obtenga ni devuelva ningún valor. Del algoritmo propuesto, responda las siguientes preguntas:

   (a) ¿Cuál es el orden de este algoritmo?

   (b) ¿En que partes del algoritmo se puede presentar comportamiento no determinista? ¿Se puede definir una estrategia general para realizar esta decisión? ¿O es algo que depende de los datos y el problema? Justifique su respuesta.

   (c) ¿Que propiedad de la pila justifica el comportamiento de búsqueda en profundidad?.

   (d) Enuncie una versión recursiva del algoritmo de búsqueda en profundidad.

   (e) ¿Cuáles son las ventajas de la versión recursiva del algoritmo? ¿Y de la versión iterativa?

   (f) Durante la ejecución de DFS, los lados que se consiguen puede conllevar una propiedad del grafo.

6. Enuncie el algoritmo de Búsqueda en Amplitud (BFS, del inglés: *breadth-first search*). Para este enunciado, solamente indique como realizar el recorrido, mas no obtenga ni devuelva ningún valor. Del algoritmo propuesto, responda las siguientes preguntas:

   (a) ¿Cuál es el orden de este algoritmo?

(b) ¿En que partes del algoritmo se puede presentar comportamiento no determinista? ¿Se puede definir una estrategia general para realizar esta decisión? ¿O es algo que depende de los datos y el problema? Justifique su respuesta.

(c) ¿Que propiedad de la pila justifica el comportamiento de búsqueda en amplitud?.

(d) Una versión recursiva del algoritmo de búsqueda en profundidad no es tan directo de implementar como una versión iterativa de éste, a diferencia de DFS en que si se puede implementar una versión recursiva. Justifique este enunciado.

(e) ¿Qué propiedad cumplen los caminos obtenidos de BFS? ¿Por qué?

7. Considere la ejecución de los algoritmos de DFS y BFS sobre grafos infinitos:

   (a) ¿Se puede asegurar que DFS terminará?. En el caso que su respuesta sea negativa, justifique y dé una condición que debe presentar el grafo para garantizar su terminación.

   (b) ¿Se puede asegurar que BFS terminará?. En el caso que su respuesta sea negativa, justifique y dé una condición que debe presentar el grafo para garantizar su terminación.

8. Proponga un algoritmo que envuelva a DFS/BFS para que garantice que el recorrido del algoritmo principal pueda realizarse para **todo** el grafo.

   (a) ¿En que partes del algoritmo se puede presentar comportamiento no determinista? ¿Se puede definir una estrategia general para realizar esta decisión? ¿O es algo que depende de los datos y el problema? Justifique su respuesta.

9. Responda las siguientes preguntas:

   (a) ¿Cuáles son los cuatro componentes necesarios para resolver un problema usando la estrategia de *backtracking*?

   (b) Postule el esquema general de resolución de problemas usando *backtracking*.

   (c) Si se ve *backtracking* como un algoritmo de grafo, ¿a qué algoritmo se asemeja? ¿Qué limitaciones trae el esquema de *backtracking* vs. el algoritmo general? ¿Por qué?

   (d) ¿Cuándo se puede utilizar un grafo de manera implícita? ¿Qué se necesita?

2.2 Ejercicios de Algoritmos sobre Grafos

1. Proponga un algoritmo que devuelva, para un grafo $G = (V, E)$, el grafo que se construye a partir del recorrido que DFS/BFS realizó. Postule dos versiones, una que devuelva un arreglo de ancestros inmediatos para los nodos que éste visitó y otro que etiquete a los nodos con su ancestro correspondiente.

   - **IMPORTANTE:** Note que este es el resultado principal con el que se va a trabajar DFS/BFS. Cuando se solicite ejecutar DFS/BFS, el resultado que debe devolver es el obtenido por este algoritmo.

2. Dado un grafo G, proponga un algoritmo que calcule las componentes conexas de un grafo. Su algoritmo debe correr en orden $O(|V| + |E|)$.

3. Dado un grafo G, proponga un algoritmo que determine si un grafo es bi-coloreable. Su algoritmo debe correr en orden $O(|V| + |E|)$.

4. Dado un grafo $G = (V, E)$, proponga un algoritmo que permita detectar si G tiene ciclos simples. Postule dos versiones: una que simplemente diga si el grafo tiene un ciclo o no, y otra que indique que ciclo encontró. Su algoritmo debe correr en orden $O(|V| + |E|)$.

5. Dado un grafo G, proponga un algoritmo que devuelva un ordenamiento de los nodos de un grafo. Un ordenamiento es una secuencia ordenada de nodos, y un ordenamiento a partir de un recorrido es un orden que se obtiene de aplicar el recorrido a un grafo. Proponga la obtención de los siguientes ordenamientos:

   (a) Ordenamiento pre-orden de DFS.

   (b) Ordenamiento post-orden de DFS. *(Use la versión recursiva de DFS.)*

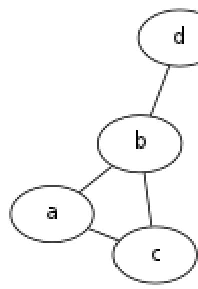
(c) Ordenamiento pre-orden inverso de DFS.

(d) Ordenamiento pre-orden de BFS.

6. Dado un grafo dirigido G, escriba el algoritmo de Kosaraju para detectar las componentes fuertemente conexas y justifique porque el algoritmo sirve para encontrarlas. *(Lea las páginas 584-590 del Sedgewick).*

7. Dado un grafo dirigido G y sus componentes fuertemente conexas, se puede construir un grafo acíclico dirigido que representa como se conectan las componentes fuertemente conexas. Proponga un algoritmo que construya dicho grafo.

8. Una componente biconexa en un grafo no dirigido $G = (V, E)$ es un subgrafo maximal tal que no existan puntos de articulación. Un **punto de articulación** es un vértice tal que la eliminación de éste en el grafo aumentaría el número de componentes conexas. Por ejemplo, en el siguiente grafo:



tiene al nodo b como punto de articulación y las componentes biconexas son $\{a, b, c\}$ y $\{b, d\}$ (observe que b pertenece a **ambas** componentes biconexas).

(a) Proponga un algoritmo que determine las componentes biconexas de un grafo G. El orden de este algoritmo es $O(|V| \cdot (|V| + |E|))$.

(b) Proponga un algoritmo que determine las componentes biconexas de G que corra en $O(|V| + |E|)$.

(c) A partir de las componentes biconexas se puede construir un grafo que represente las conexiones entre las componentes biconexas. ¿Cómo se puede realizar dicho grafo?.

9. Dado un grafo no dirigido $G = (V, E)$, postule un algoritmo que determine si existe exactemente un camino simple entre un par de vértices de V. Su algoritmo debe correr en orden $O(|V| + |E|)$.

10. Dado un grafo dirigido $G = (V, E)$ tal que el grado externo de cada nodo es uno. De un algoritmo que, dado un vértice $v \in V$, determine el número de nodos que se necesitan cruzar de v al inicio de un ciclo.

11. Dado un grafo dirigido bipartito. Proponga un algoritmo que determine si es acíclico y de serlo, devolver todos los nodos fuentes y nodos sumideros. Su algoritmo debe poder dar un resultado en orden de tiempo $O(|V| + |E|)$. *(Un nodo fuente es aquél que no tiene predecesores mientras que un nodo sumidero es aquél que no tiene sucesores.)*

12. Para un grafo (dirigido o no) $G = (V, E)$:

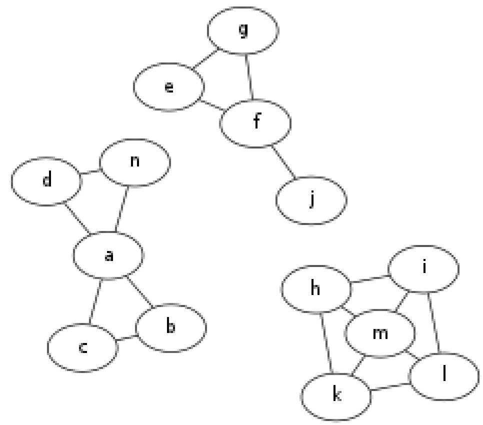
(a) Proponga un algoritmo que calcule la matriz de alcance. Su algoritmo debe correr en tiempo $O(|V| \cdot (|V| + |E|))$. *(La matriz de alcance es símilar a la matriz de adyacencia, pero en vez de representar la relación de adyacencia, representa la relación de alcanzabilidad.)*

(b) Proponga un algoritmo que calcule, para todo par de nodos, cual es el tamaño del camino más corto que conecta ese par de nodos. El resultado debe ser símilar a la matriz de alcance, pero en vez de tener valores booleanos, tendrá el tamaño del camino más corto entre ese par (-1 si los nodos no son alcanzables).

13. Un ciclo hamiltoniano es un ciclo tal que puede pasar por todos los nodos una sola vez (exceptuando el nodo de inicio/final). Proponga un algoritmo que detecte la existencia de ciclos hamiltonianos en un grafo. Su algoritmo debe correr en tiempo $O(|V|^{|V|})$. *(Considere la construcción de un grafo de caminos para resolver el problema.)*

2.3 Ejercicios de Corridas de Algoritmos.

1. Para el siguiente grafo:



calcule, utilizando el correspondiente algoritmo:

(a) DFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el. *(Primero aplicar DFS sobre a, luego si no fue visitado, sobre b, c,..., hasta llegar al último nodo.)*

(b) BFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el BFS como para el sub-criterio de selección en el.

(c) Ordenamiento pre-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.

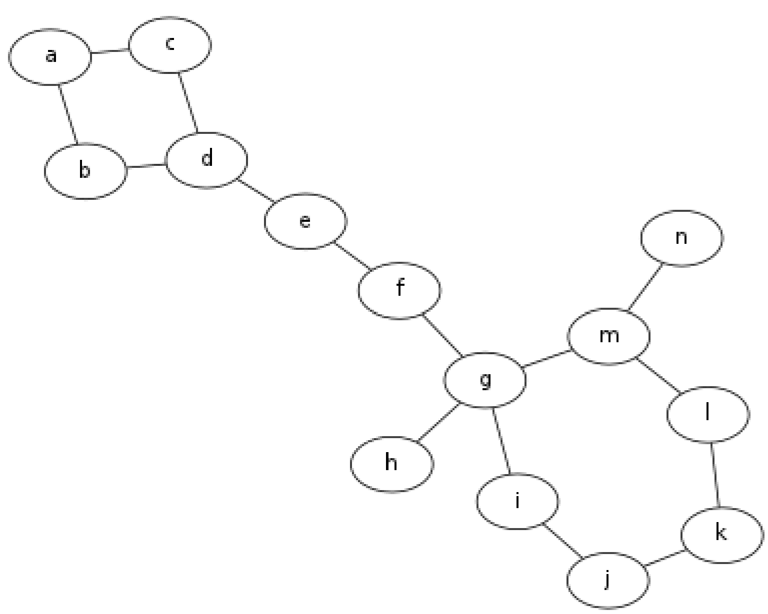
(d) Ordenamiento post-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.

(e) Las componentes conexas de este grafo.

(f) Si existen ciclos simples sobre este grafo.

(g) Si el grafo es bipartito.

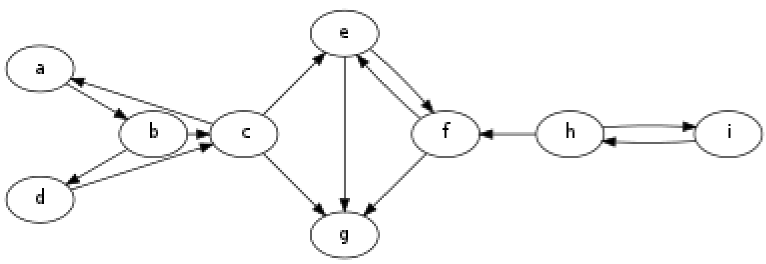
2. Para el siguiente grafo:



calular:

(a) DFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(b) BFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el BFS como para el sub-criterio de selección en el.
(c) Ordenamiento pre-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(d) Ordenamiento post-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(e) Las componentes conexas de este grafo.
(f) Si existen ciclos simples sobre este grafo.
(g) Si el grafo es bipartito.

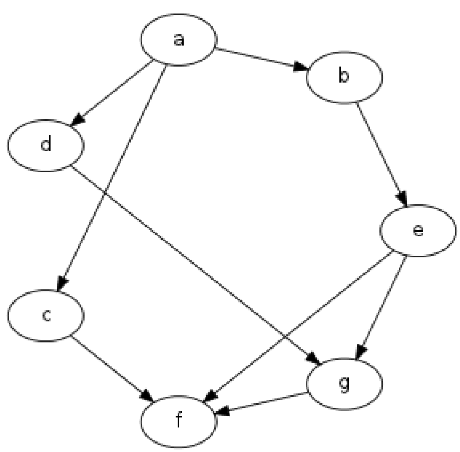
3. Para el siguiente grafo dirigido:



calcular:

(a) DFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(b) BFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el BFS como para el sub-criterio de selección en el.
(c) Ordenamiento pre-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(d) Ordenamiento post-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.
(e) Las componentes fuertemente conexas de este grafo.
(f) Si existen ciclos simples sobre este grafo.

4. Para el siguiente grafo dirigido:



calcular:

(a) DFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.

(b) BFS sobre todos los nodos. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el BFS como para el sub-criterio de selección en el.

(c) Ordenamiento pre-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.

(d) Ordenamiento post-orden DFS. Aplique el algoritmo en orden alfabético para seleccionar tanto los nodos en el DFS como para el sub-criterio de selección en el.

(e) Las componentes fuertemente conexas de este grafo.

(f) Si existen ciclos simples sobre este grafo.

3 Caminos de Costo Mínimo

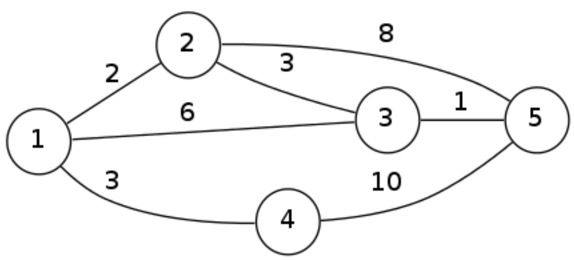
1. Postule el algoritmo de Dijkstra. Su algoritmo debe devolver el grafo obtenido de aplicar el recorrido. Postule dos versiones, una que devuelva un arreglo de ancestros inmediatos para los nodos que éste visitó y otro que etiquete a los nodos con su ancestro correspondiente.

2. ¿Cuál es el orden de ejecución del algoritmo de Dijkstra? Considere distintas formas de implantar una cola de prioridades para el cálculo del orden de ejecución.

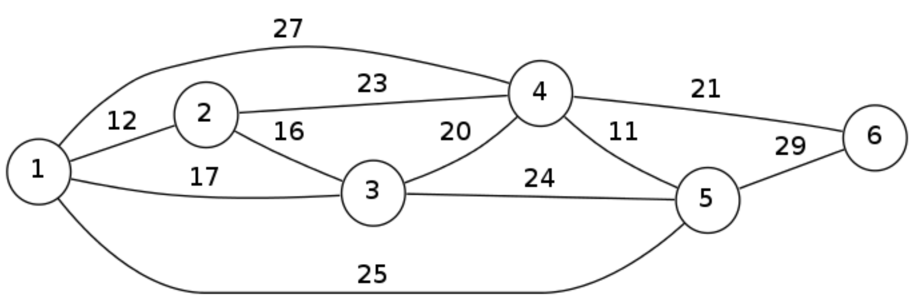
3. Considere un grafo en el que todos los lados tienen como costo una misma constante (por ejemplo, 1). ¿A qué algoritmo se asemeja Dijkstra cuando se ejecuta con este grafo?

4. Ejecute el algoritmo de Dijkstra para los siguientes grafos. Aplique el algoritmo en orden numérico para seleccionar tanto los nodos en el proceso de encolado como para el sub-criterio de selección en el.

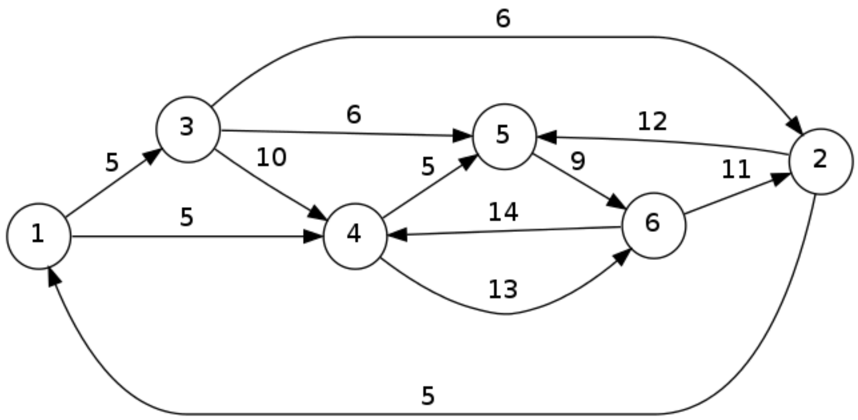
(a)



(b)



(c)



4 Resolución de problemas.

Para cada uno de los siguientes problemas, proponga un grafo que modele el problema, indique que problema va a resolver y proponga un algoritmo que resuelva el problema:

1. Considere un mapa de islas organizado en una matriz. Dicha matriz tiene como posibles valores 0 (correspondientes a una región de agua) y 1 (correspondientes a una región de tierra). Se quiere determinar el área de cada isla, su perímetro externo, las islas que tienen lagos y de estas su perímetro interno.

   (a) **Versión díficil:** Determine si puede hacer este cálculo en un solo pase.

2. Una reina quiere saber si su palacio es lo suficientemente interesante como para poder variar la rutina de vez en cuando. Para eso, quiere saber si llegar de un punto a otro en el palacio es divertido, siendo su definición de divertido que haya mas de una manera de poder llegar de ese punto a otro sin cruzar dos veces por un punto. Dado un palacio y dos puntos del palacio, determine si el recorrido de ambos puntos es divertido.

3. Considere un pueblo en el que hay una central telefónica ubicada en un punto cualquiera del pueblo y repetidores telefónicos regados en éste. Tiene un mapa del pueblo en que aparecen todas las casas, los repetidores y la central telefónica y le pide a ud. que indique como ofrecer servicio telefónico a todas las casas usando el menor número de conexiones considerando que la central no puede ofrecer el servicio directamente a una casa sino que tiene que pasar a juro por un repetidor.

4. Tres misioneros y tres caníbales deben cruzar un río usando un bote que pueden llevar a lo sumo dos personas. En cada lado del río, si hay misioneros, éstos deben superar o igualar a los caníbales en número (si hay más, los caníbales se comen a los misioneros) y además el bote no se puede mover por si solo. ¿Existe alguna forma de poder mover a los misioneros y los caníbales para que puedan llegar al otro lado del río sin que no hayan pérdidas?. De existir esta forma, se quiere saber en cuantos pasos se puede llegar a ese punto en el menor número de pasos posibles. *(Se pide cuantos pasos, no cuales.)*

5. Considere el *puzzle* de las Torres de Hanoi: tres torres con N discos de radio creciente colocados en la primera torre, se desea llevar los discos de la primera a la tercera torre de tal forma que en cualquier momento siempre se respete la condición de que un disco puede estar montado o en el fondo de la torre o por encima de otro disco de radio superior. Determine un algoritmo que determine cuales pasos se necesitan para resolver el rompecabezas en el menor número de pasos. Darse cuenta de que el mínimo número de pasos es $2^n - 1$.

6. Proponga un algoritmo que devuelva una configuración de fichas de dominó de tal forma que se utilicen todas las fichas, realizando conexiones de forma correcta y de tal forma que ambos extremos tengan el mismo valor.

7. En una empresa, se quiere agrupar la gente por confianza. Se hizo una encuesta y se determinó por cada empleado, que compañeros de ellos confían. Además, se espera que si un empleado A confía en un empleado B, y un empleado B confía en un empleado C, A confiará en C. Se desea separar los empleados tal que haya el menor número de grupos que cumpla con que todos los empleados de un grupo tengan confianza, sea directa o indirectamente.

8. Proponga un algoritmo que, dado un tablero del juego de La Vieja, saber si existe alguna secuencia de jugadas que permita que el jugador X gane al jugador O.

9. Considere un lenguaje de programación (con sintaxis similar a Java) en el que solo tenga asignación, condicionales de dos posibilidades (el else es obligatorio) y una instrucción de impresion. Una variable es considerara inútil si dicha variable no aparece en un lugar en que sea necesitada (que aparezca en una expresión o en un print). Proponga un algoritmo que consiga las variables inútiles del programa.

10. Un dibujo se puede representar como una matriz de píxeles. Proponga un algoritmo que permita hacer la operación de llenado - reemplazar un área común de un mismo color por otro color. (El dibujo es el grafo.)

11. Risk es un juego en el cual varios jugadores intentan conquistar el mundo. El tablero del juego es un mapa dividido en países o regiones hipotéticas. Durante un turno, el ejército instalado en una región solo puede atacar aquellos países que comparten una frontera. Al conquistar el país, se pueden mover tropas a dicho país.

12. Durante una partida, un jugador comienza una secuencia de conquistas hasta llegar a la capital del contrincante. Típicamente, el jugador elige países de tal manera que se pueda llegar a la capital enemiga en el menor número de países. ¿Cómo determinar dicho número?

13. Proponga un algoritmo que resuelva el problema de las 8 reinas: se tiene un tablero de ajedrez (de tamaño 8×8) y 8 reinas. ¿Cómo se pueden ubicar las 8 reinas de tal forma que ninguna se amenace?. De la misma forma, ¿se pueden ubicar las 8 reinas de tal forma que todas estén amenazadas simultáneamente.

    (a) ¿Qué cambios haría a su algoritmo para poder atacar el problema general?

14. Proponga un algoritmo que, dado un sitio web en el que todas las páginas pueden ser alcanzadas desde una página principal, determine el promedio del camino más corto (en número de páginas visitadas) entre dos nodos cualesquiera. (Es el promedio de **todos** los caminos más cortos.)

15. Un paso de edición es una transformación de una palabra x a una palabra y, ambas pertenecientes a un diccionario tal que x puede ser transformada en y añadiendo, eliminando o cambiando una letra (por ejemplo, casa $\Rightarrow$ cosa, pero $\Rightarrow$ perro y salsa $\Rightarrow$ sala) y, además, x aparece antes que y según el orden estípulado en los diccionarios.

    Una secuencia de edición es una secuencia de transformaciones es una secuencia de palabras $w_1, w_2, \ldots, w_n$ tal que la transformación de w_i a w_{i+1} es un paso de edición para todo i de 1 a $n-1$. (Observe que, al estar ordenadas lexicográficamente, no puede haber palabras repetidas.)

    Proponga un algoritmo que determine la secuencia de edición más larga en un diccionario dado.

16. En el país de Distantistán, su capital se compone por calles e intersecciones. Cada calle conecta a 2 intersecciones. El shah de Distantistán quiere ubicar guardias en algunas intersecciones tal que todas las calles e intersecciones estén seguras. Un guardia en una intersección puede resguardar todas las calles e intersecciones que estén adyacentes a su intersección asignada... pero lamentablemente los guardias no se llevan bien. Si dos guardias cubren una calle, problemas habrán y peleas se formarán. Dado un mapa de las calles de la capital de Distantistán, proponga un algoritmo que suministre al shah el mínimo número de guardias que necesita para cubrir todas las calles e intersecciones de la capital real.

17. Considere un sistema de transporte en masa compuesto por líneas y estaciones. Una línea está compuesta por una secuencia de estaciones y una estación puede hacer conexión a otra estación en otra línea. Ahora, considere que viajar entre estaciones de la misma línea toma 1 unidad de tiempo en recorrerse, mientras que las conexiones tienen 3 unidades de tiempo. Dadas dos estaciones, ¿cómo se calcula el menor tiempo entre ambas estaciones?

18. Considere un laberinto numérico representado como un arreglo bi-dimensional de números comprendidos entre 0 y 9. El laberinto puede ser recorrido cualquier dirección ortogonal (norte, sur, este y oeste). Considerando que cada celda representa un costo, encuentre la manera más barata de llegar de la esquina superior izquierda a la esquina superior derecha.

19. Considere un rascacielos de 100 pisos, numerados cada uno del 0 al 99. Tiene n ascensores ($1 \leq n \leq 5$) que viajan a velocidades distintas. Para cada i en $\{1, 2, \ldots, n\}$, el ascensor i se toma T_i segundos en viajar entre dos pisos adyacentes sin importar la dirección *(Por ejemplo, pasar del piso 2 al 3 se toma T_i segundos, sin*

importar que el ascensor i se pare en 2 o en 3). Los ascensores no necesariamente se paran en todos los pisos. Además, para complicar el asunto, no todos los pisos son accesibles por un ascensor.

Ud. quiere ver como puedes llegar del piso 0 al piso k lo más rápido posible suponiendo que, al inicio, todos los ascensores que conectan al piso 0 están disponibles y (por simplicidad) la operación de llamar a un ascensor en cualquier piso toma exactamente un minuto (por supuesto, tanto el ascensor en que se llegó al piso como el ascensor a tomar deben parar en el mismo piso). Además, por un pequeño detalle de diseño, no hay escaleras en el rascacielos. Puede además, suponer que nadie va a tomar el ascensor mientras ud. lo toma, por lo que no tiene que considerar las paradas.

Proponga un algoritmo que permita calcular el menor número de segundos requeridos para ir del piso 0 al piso k. Pasar por el piso k sin bajarse en el (p.e. montado en un ascensor que no se para en ese piso) no cuenta como "ir al piso k".

20. Una ciudad es cubierta por un número de estaciones de bomberos. Algunos residentes se quejan de que la distancia de sus hogares a la estación más cercana es muy lejana, por lo que se decide construir una nueva estación. Suponga que los hogares y las estaciones están ubicadas en intersecciones y que al menos hay un hogar por interesección. Dada la lista de las intersecciones donde se encuentran las estaciones de bomberos ya construídas, proponga un algoritmo que indique donde ubicar la nueva estación de tal forma que reduzca lo más posible la distancia de todas las casas a su estación cercana.

21. Suponga que hay n servidores de correo conectados por cables de red. Cada uno de los m cables conecta dos computadoras y cada cable tiene su propia medida de latencia (cuanto tiempo toma en transferir algo) medida en milisegundos usados para enviar un correo. ¿Cuál es el menor tiempo para enviar un mensaje de un servidor S a un servidor T, estando ambos conectados? Suponga que no existe retraso alguno en los servidores.

22. Una empresa de transporte envía bienes de una ciudad a otra. La empresa ha llegado a un acuerdo con una cadena de hoteles para que sus choferes puedan hospedarse en los hoteles de manera gratuita. Los choferes solo pueden estar en ruta hasta 10 horas por día. La empresa quiere buscar una ruta desde el origen al destino de tal forma que un chofer tome máximo 10 horas entre una sucursal del hotel y otra, buscando minimizar el número de días que tome para llegar al destino.

23. Dos personas, A y B, son muy buenos amigos. Ellos piensan ir al estadio a ver un *partido de algún deporte o un evento de su preferencia* y al terminar el partido/evento, ellos tienen que volver a sus casas. Como desean conversar del evento lo más posible, pero a la vez quieren llegar pronto a sus casas, desean saber que ruta pueden tomar para que puedan llegar lo más rápido posible a sus casas pero que estén el máximo de tiempo posible juntos. (Suponga que tiene un mapa en que las calles se miden por el tiempo que toma cruzarla.)

Referencias

[1] `http://en.wikipedia.org/wiki/Hierholzer%27s_algorithm`