

CC 20 A Otoño 2006

Tarea Recuperativa

Instrucciones: La siguiente tarea permite recuperar la nota por el ítem “Tareas” del curso. La tarea involucra investigar la solución, para lo cual se puede usar biblioteca, Internet, etc. La tarea es individual. El informe deberá entregarse en Secretaría Docente del DCC (Blanco Encalada 2120, cuarto piso) hasta el 16 de agosto 2006 a las 17 hrs. Exigir un recibo. No se aceptarán entregas posteriores. Salvo que el Profesor del curso lo autorice expresamente, la nota por esta tarea será “aprobado” o “reprobado”, y en el cálculo de la nota final del curso se pondrá 4.0 en la componente “Tareas” para propósito de calcular la nota final.

Tema: Conectividad en redes. Este es un tema muy interesante para muchas empresas, por ejemplo, en la distribución y transmisión de electricidad.

Antecedentes.

Un grafo no dirigido es conexo si hay una trayectoria de cualquier vértice a cualquier otro vértice. Un grafo es **biconexo** si hay al menos dos trayectorias disjuntas de cada vértice a cualquier otro vértice. Se considera también que un grafo conexo con un solo arco es biconexo.

Los grafos biconexos tienen una muy buena conectividad: si por alguna razón, una de las trayectorias que conectan dos vértices, no puede usarse, todavía los vértices quedan conectados usando la otra trayectoria. (Piense lo útil que es esto cuando hay desastres naturales que afecten líneas de transmisión eléctrica, caminos, etc.).

La teoría dice que si un grafo no es biconexo, entonces puede ser particionado en subgrafos, cada uno de los cuales es biconexo. Este es un resultado muy importante.

Problema

Dado un grafo no dirigido cualquiera, buscar un algoritmo que encuentre los componentes biconexos de ese grafo. Programar el algoritmo y aplicarlo a un ejemplo específico.

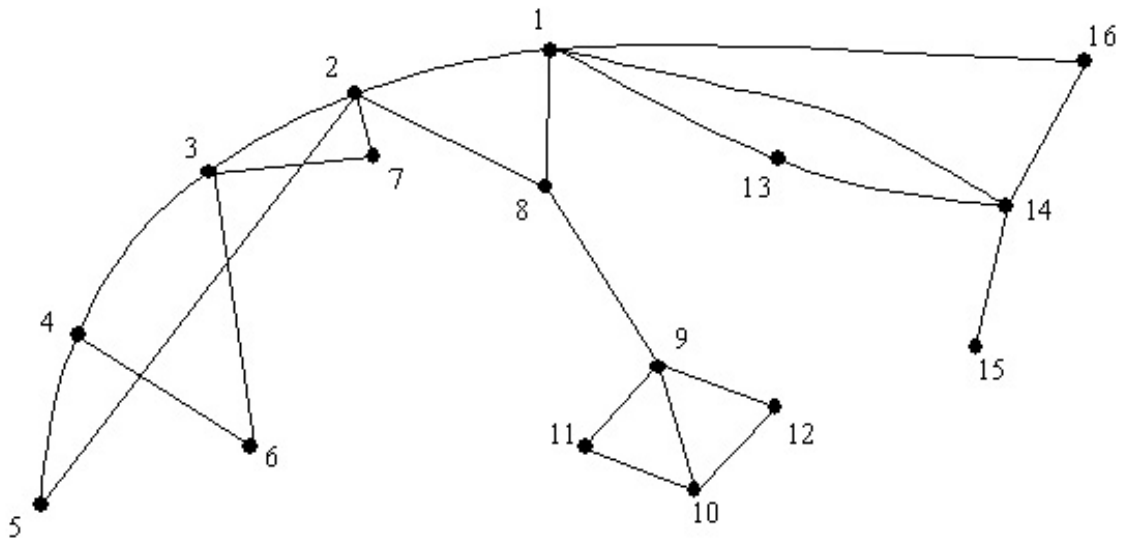
Entregables

Se pide entregar lo siguiente:

1. Portada, con nombre del alumno
2. Explicación legible del método a usar, definiciones a emplear, etc.
3. Algoritmo utilizado
4. Referencia a la fuente bibliográfica utilizada (por ej., nombre autor, título del artículo, journal, volumen, No., año, págs.)

5. Programa, escrito en cualquier lenguaje de programación, del algoritmo
6. Explicación del programa (estructuras de datos utilizadas, variables).
7. Resultados arrojados por el programa al caso ejemplo, y su interpretación (esto último ilustrado con diagramas).

Caso ejemplo



donde:

1: Santiago, 2: Melipilla, 3: San Antonio, 4: cartagena, 5: Valparaiso, 6: Quilpué, 7: Viña del Mar, 8: San Bernardo, 9: Rancagua, 10: Rengo, 11: Las Cabras, 12: San Fernando, 13: San Felipe, 14: Los Andes, 15: Mendoza, 16: Rinconada.