PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Laboratorio 5

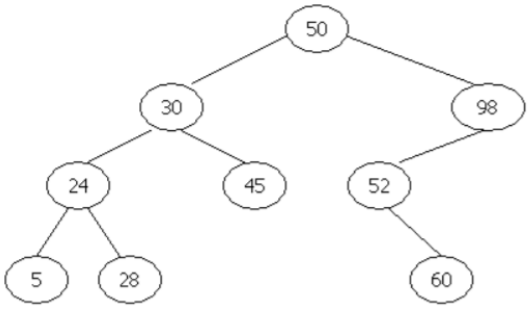
2017-1

Indicaciones generales:

- Duración: 2h 50 min.
- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Si la implementación es significativamente diferente a la estrategia indicada o no la incluye, la pregunta será corregida sobre el 50% del puntaje asignado y sin derecho a reclamo.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 60% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Su trabajo deberá ser subido a PAIDEIA en el espacio indicado por los jefes de práctica.

Pregunta 1 (10 puntos) ***Binary Search Tree Traversals***

En el curso hemos aprendido las propiedades de un árbol binario de búsqueda y los diferentes recorridos que se pueden realizar. En este problema se tiene que implementar un programa que, dado el orden de visita de un árbol en pre-orden calcule el orden de visita en post-orden. Por ejemplo, considere el siguiente árbol:



Sus recorridos serían:

- Pre-orden: 50 30 24 5 28 45 98 52 60
- Post-orden: 5 28 24 45 30 60 52 98 50

Este problema consiste en calcular el recorrido en post-orden, dado el recorrido en pre-orden.

Entrada

El primer número es un número entero T, el número de nodos,

A continuación viene la secuencia del recorrido en pre-orden separada por espacios

Salida

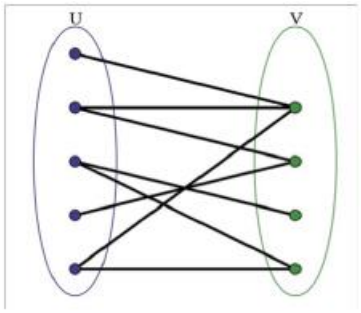
El recorrido en post-orden, separado por espacios

Ejemplo:

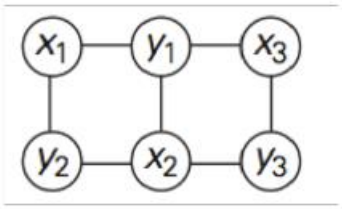
Entrada	Salida
9 50 30 24 5 28 45 98 52 60	5 28 24 45 30 60 52 98 50

Pregunta 2 (10 puntos) Grafo Bipartito

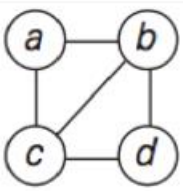
Un grafo es bipartito si sus vértices pueden ser divididos en dos conjuntos independientes, U y V, de tal forma que cada arista (u, v) o conecta un vértice de U con uno de V o un vértice de V con uno de U. En otras palabras, por cada arista (u, v), o u pertenece a U y v a V, o u pertenece a V y v a U. También se puede decir que no existe alguna arista que conecte vértices del mismo conjunto. En la siguiente figura, los grafos I y II son bipartitos pero el grafo III no lo es.



(I)



(II)



(III)

Implemente un programa en C que determine si un grafo es bipartito o no. En caso de ser bipartito, el programa debe mostrar los elementos que pertenecen a cada conjunto. Puede ser la representación de grafo que desee.

Entrada

Para cada caso de prueba, la primera línea contiene 2 números: "v", que representa el número de vértices y "e", que representa el número de aristas. La siguiente línea contiene "v" elementos que representan los vértices. Las siguientes "e" líneas representan las aristas a crear entre cada par de vértices. Nótese que el grafo con el cual se debe trabajar es un grafo no dirigido.

Salida

En caso no se puedan generar los 2 conjuntos, se debe mostrar el mensaje "No es bipartito". Caso contrario, se deben mostrar los elementos de los 2 conjuntos generados.

Ejemplo:

Entrada	Salida
4 5 1 2 3 4 1 2 1 3 2 3 2 4 3 4	No es bipartito
6 7 1 2 3 4 5 6 1 2 1 4 2 3 2 5 3 6 4 5 5 6	1 3 5 2 4 6

Profesores del curso: Marco Sobrevilla e Ivan Sipiran

Pando, 24 de junio del 2017