PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Segundo Examen
(Segundo Semestre de 2017)

Horario 0581: prof. Andrés Melgar
Horario 0582: prof. Iván Sipirán

Duración: 3 horas
Nota:

- No se permite el uso de material de consulta.
- **Debe utilizar comentarios para explicar la lógica seguida en los programas elaborados, así como nombres de variables apropiados.**
- La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

Cuestionario:

PARTE PRÁCTICA

Pregunta 1 (7 puntos) FrontBackSplit - Linked List Problems by Nick Parlante (Traducción Libre) Se pide implementar la función `FrontBackSplit()` que dada una lista, la divida en dos sub listas: una para la mitad frontal y otra para la mitad posterior. Si el número de elementos es impar, el elemento adicional debe ir en la lista frontal. De esta forma `FrontBackSplit()` en la lista $\{2, 3, 5, 7, 11\}$ debería arrojar las listas $\{2, 3, 5\}$ y $\{7, 11\}$. Hacer esto bien para todos los casos es más difícil de lo que parece. Debe verificar su solución contra algunos casos (longitud = 2, longitud = 3, longitud = 4) para asegurarse de que la lista se divida correctamente cerca de las condiciones de límite. Si funciona bien para una longitud de 4, probablemente funcione bien para una longitud de 1000. Es probable que necesite un código de caso especial para tratar los casos (longitud < 2).

Nota. Probablemente, la estrategia más simple es calcular la longitud de la lista, luego usar un bucle `for` para saltar sobre el número correcto de nodos para encontrar el último nodo de la mitad frontal, y luego cortar la lista en ese punto. Hay una técnica que utiliza dos punteros para recorrer la lista. Un puntero "lento" avanza un nodo a la vez, mientras que el puntero "rápido" va dos nodos a la vez. Cuando el puntero rápido llega al final, el puntero lento estará a medio camino. Para cualquiera de las estrategias, se requiere cuidado para dividir la lista en el punto correcto.

Nota 2. Su solución deberá venir acompañado de un programa principal en donde se prueben los siguientes casos: lista de tamaño nulo, lista de tamaña 1, lista de tamaño 2, lista de tamaño 3, lista de tamaño 4 y lista de tamaño 5. Para cada caso deberá imprimir la lista original y las listas obtenidas.

Pregunta 2 (7 puntos) Se tiene un grafo dirigido con N vértices y M aristas. Se dice que un vértice X es padre de un vértice Y si existe la arista de X a Y. El padre maestro es el vértice que no tiene padre y puede tener 0 o más hijos. Se dice que un vértice es feliz si tiene más hijos que su padre. Contar el número de vértices felices en el grafo dado. El grafo no tiene ciclos o bucles. **Sugerencia:** usar búsqueda en profundidad como base para recorrer el grafo ordenadamente.

Entrada: La primera línea consiste de dos enteros separados por espacios denotando N y M y a continuación M líneasque contienen dos enteros separados por espacios X y Y denotando la existencia de

una arista entre los vértices X y Y.

Salida: Imprima el número de vértices felices en el grafo

Ejemplo de Entrada:

```
4 3
1 2
2 3
2 4
```

Ejemplo de Salida:

```
1
```

PARTE ELECTIVA

Para **UNA** de las preguntas que se presentan a continuación, elabore un programa en C que resuelva el problema descrito.

Pregunta 3 (6 puntos) Dada una lista simplemente enlazada de números enteros, se pide que implemente una función que convierta la lista en un árbol binario de búsqueda balanceado. Un árbol binario de búsqueda balanceado es aquel árbol en que la diferencia de sus alturas de su sub árbol izquierdo y su sub árbol derecho es a lo más uno y tanto el sub árbol izquierdo como el sub árbol derecho son árboles binarios de búsquedas balanceados.

Nota. Su solución deberá venir acompañado de un programa principal en donde se imprime la lista usada y también se recorra el árbol generado, imprimiendo la raíz usando el recorrido en orden.

Pregunta 4 (6 puntos) Implementar un programa en C que determine si un grafo dado es un árbol. La información que se tiene disponible es el número de vértices N y el grado de cada vértice. El grado de un vértice es el número de aristas conectadas al vértice.
Entrada: La primera línea contiene un entero N, el número de vértices. La segunda línea contiene N enteros separados por espacio, los grados de los N vértices

Salida: Imprima SI si e grafo es un árbol o NO, en caso contrario

Ejemplo de Entrada:

```
3
1 2 1
```

Ejemplo de Salida:

```
SI
```

Profesores del curso: Andrés Melgar
Iván Sipiran

Pando, 9 de diciembre de 2017