

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Examen 2

(Primer semestre 2016)

Indicaciones generales:

- Duración: 3h.
- Materiales o equipos a utilizar: Apuntes de clase personales escritos a mano.
- 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 o si la implementación es significativamente diferente a la estrategia indicada, el alumno no obtendrá el puntaje completo en dicha pregunta.
- 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 considerado dentro de la evaluación.
- Cada programa debe ser guardado y subido a PAIDEIA en el espacio indicado por los Jefes de Práctica.
- **La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Cuestionario:

Pregunta 1 (1.5 puntos)

Una tabla hash de longitud 10 usa *open addressing (linear probing)* con función hash $h(k) = k \bmod 10$. Luego de insertar 6 valores en una tabla hash vacía, la tabla es la que se muestra a continuación:

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

¿Cuál de las siguientes opciones indica un posible orden en que las llaves pueden haber sido insertadas en la tabla? Justifique su selección.

- a) 46, 42, 34, 52, 23, 33
- b) 34, 42, 23, 52, 33, 46
- c) 46, 34, 42, 23, 52, 33
- d) 42, 46, 33, 23, 34, 52

Pregunta 2 (1.5 puntos)

Considere el siguiente código:

```
// Un nodo del ABB
typedef struct node {
    int data;
    struct node *left, *right;
}TNode;

int count = 0;

void print(TNode *root, int k)
{
    if (root != NULL && count <= k)
    {
        print(root->right, k);
        count++;
        if (count == k)
            printf("%d ", root->data);
        print(root->left, k);
    }
}
```

¿Qué realiza la función *print()*? Esta función recibe como parámetros la raíz de un Árbol Binario de Búsqueda (ABB) y un entero positivo k . Justifique su respuesta.

- a) Imprime el k -ésimo menor elemento del ABB
- b) Imprime el k -ésimo mayor elemento del ABB
- c) Imprime el nodo más a la izquierda del nivel k contado desde la raíz
- d) Imprime el nodo más a la derecha del nivel k contado desde la raíz

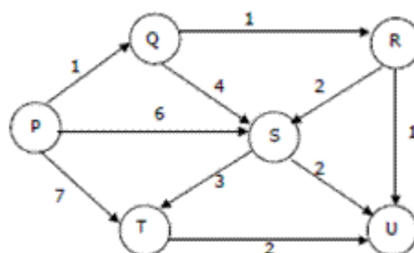
Pregunta 3 (2 puntos)

Si realizamos las dos siguientes modificaciones al algoritmo de Dijkstra, ¿podría ser usado para encontrar el camino más largo? Asuma que el grafo no posee ciclos. Justifique su respuesta.

- Inicializar todas las distancias con “menos infinito” en lugar de “más infinito”.
- Modificar la condición del algoritmo para actualizar la distancia de un vértice adyacente v del vértice u que actualmente se está considerando, solo si $dist[u] + graph[u][v] > dist[v]$. La versión original de algoritmo considera el signo de comparación opuesto.

Pregunta 4 (3 puntos)

Suponga que se ejecuta el algoritmo de caminos mínimos de Dijkstra en el siguiente grafo dirigido, considerando a P como el vértice inicial. ¿En cuál orden son incluidos los nodos en el conjunto de vértices para los cuales ya se conoce cuál es su camino más corto desde el vértice inicial? Justifique su respuesta.



Pregunta 5 (7 puntos)

Junto con tus amigos han decidido jugar a la ruleta rusa. Para iniciar el juego, el grupo se ha dispuesto en una mesa circular. El juego consiste de las siguientes reglas:

- Todos los integrantes han hecho una apuesta monetaria.
- El juego empieza en un determinado miembro del grupo, el cual sostiene una tarjeta que indica de quién es el turno.
- Antes de iniciar el juego, se escoge un número entero k , el cual indica la cantidad de veces que la tarjeta se pasará hacia el compañero de la derecha.
- Después de que la tarjeta ha sido pasada k veces, el portador de la tarjeta pierde y tiene que retirarse de la mesa. En este caso, la tarjeta pasa al compañero de la derecha y el juego empieza de nuevo usando el mismo k . Recuerde que ahora hay un miembro menos en la mesa.
- El juego finaliza cuando queda un solo amigo en la mesa, quien se lleva el botín de la apuesta.

Tú te has dado cuenta que puedes emplear tus conocimientos de estructuras de datos para implementar un programa que use listas enlazadas circulares para resolver el problema y saber quién ganará el juego, dada una configuración inicial. **Nota:** Una lista enlazada circular es aquella en la que el último elemento apunta al primero. Tomar en cuenta que cuando un amigo tiene que salir de la mesa significa que hay que borrar un nodo en la lista circular.

Entrada: El programa recibe como entrada un archivo de texto cuya primera línea contiene la cantidad de casos de prueba. A continuación, para cada caso se indica la cantidad de amigos seguido por sus nombres (uno en cada línea). Además el orden de entrada indica cómo están sentados en la mesa (el segundo en la lista de entrada está sentado a la derecha del primero y así sucesivamente). La siguiente línea indica el nombre del amigo que empieza el juego y por último una última línea con el número k . No hay líneas vacías entre cada caso.

Salida: Los nombres de los ganadores de cada caso (uno por línea).

Ejemplo de Entrada:	Ejemplo de Salida:
2 4 Felipe Elena Pedro Antonio Pedro 7 8 Luis Marco Cesar Tatiana Rodrigo Luz Esteban Diana Cesar 4	Felipe Rodrigo

Implemente un programa en C que resuelva el problema planteado.

Pregunta 6 (5 puntos)

En un grafo dirigido G se desea verificar si existe un vértice i tal que existe una arista desde cada vértice $j \neq i$ hasta i , y no existe arista alguna desde i hasta cualquier otro vértice. Implemente un programa en C que permita resolver este problema. Puede usar la representación de grafo que desee. Pruebe su programa con al menos 2 casos: uno en que exista un vértice de este tipo y otro en el que no.

Profesores del curso: Fernando Alva
Iván Sipirán

Pando, 2 de julio de 2016