PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Examen 2
(Segundo 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 de 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 en un archivo con el nombre *preg#codigo_de_alumno.c* 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

PREGUNTA 1 (7 puntos)

Se le pide implementar un editor de textos simple. Inicialmente, este editor contiene una cadena vacía *S* y se pueden realizar operaciones de los siguientes tipos:

1. *append* (*W*) - Agrega una cadena de caracteres *W* al final de *S*.

2. *delete* (*k*) - Elimina los k últimos caracteres de S.

3. *print* (*k*) - Imprime el k-ésimo caracter de *S*.

4. *undo* - Deshace la última operación (no previamente deshecha) de tipo (1) *append* o (2) *delete*, volviendo al estado en el que estaba antes de esa operación.

Entrada

La primera línea contiene un entero, *Q*, denotando el número de operaciones. Cada una de las siguientes *Q* líneas define una operación a realizarse. Cada operación comienza con un número entero *t* (*t* $\in \{1,2\ 3,\ 4\}$), denotando el tipo de operación como definido en el problema dado (1-*append*, 2-*delete*, 3-*print* y 4-*undo*). Si la operación requiere un argumento, *t* es seguido por un argumento separado por un espacio.

Salida

Cada operación 3 (*print*) debe imprimir el k-ésimo carácter de S en una nueva línea.

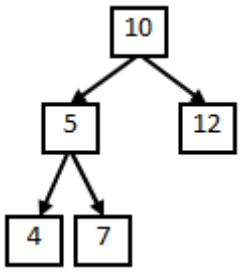
Ejemplo

8 1 abc 3 3 2 3 1 xy 3 2 4 4 3 1	c y a

Nota: Puede utilizar el código fuente base entregado en PAIDEIA.

PREGUNTA 2 (7 puntos)

En un árbol binario de búsqueda, un camino es una secuencia de nodos que van desde la raíz hasta un nodo hoja. Dado un árbol binario de búsqueda y un número *q*, implementar una función en C que imprima todos los caminos en donde la suma de todos los nodos en el camino es igual al número dado *q*. Por ejemplo en el siguiente árbol:



Si q = 22, la función debe imprimir los caminos

- 10 12
- 10 5 7

Nota: Utilice el código fuente base entregado en PAIDEIA.

PARTE ELECTIVA

Resuelva las preguntas que se presentan a continuación de tal manera que sumen como máximo **6 puntos**.

PREGUNTA 3 (3 puntos)

Una persona desea encontrar la moda de las edades de todas las personas del mundo y te contrata para que la puedas encontrar. ¿Cómo resolverías el problema de manera eficiente? ¿Usarías un arreglo o una lista enlazada? y ¿cómo la usarías? Justifica tus respuestas.

PREGUNTA 4 (3 puntos)

¿Cómo guardarías un árbol binario de búsqueda en un arreglo de manera eficiente de tal manera que luego pueda ser reconstruido? Pista: Guardar un nodo en la i-ésima posición y sus hijos en la 2i-ésima y 2i+1-ésima posición no es la forma más eficiente. Justifica tu respuesta.

PREGUNTA 5 (6 puntos) – Eliminación en Hashing

Diseñar un algoritmo que reciba como entrada una Tabla Hash *T* (con open addressing), el tamaño *m* de *T* y una clave *k*. El algoritmo debe eliminar la clave *k* de la tabla (siempre y cuando se encuentre). Tenga en cuenta que como usamos open addressing, los elementos después del elemento eliminado podrían haber sido producto de colisión. Por lo tanto, ante cada eliminación, se debe volver a aplicar hashing sobre los elementos después del eliminado para reorganizar la tabla Hash.

¿Cuál es la complejidad del algoritmo que ha diseñado?

Considere un función hash $h(k) = k \bmod m$.

Profesor del curso: Marco Sobrevilla

Iván Sipiran

San Miguel, 3 de Diciembre de 2016