CC3301 Programación de Software de Sistemas – Tarea 3 – Semestre Otoño 2022 – Prof.: Luis Mateu

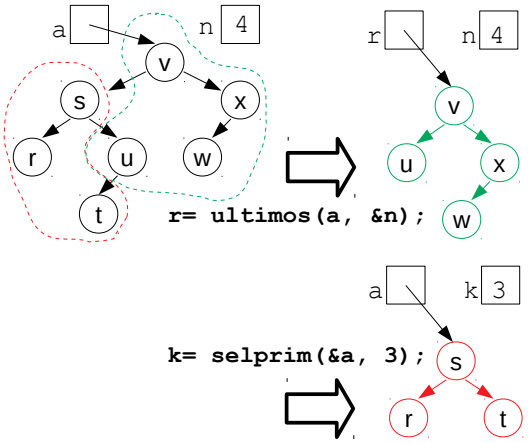
En esta tarea Ud. deberá programar 2 funciones para seleccionar partes de un arbol binario de búsqueda. Las 2 funciones pedidas son:

```
typedef struct nodo {
  int x;
  struct nodo *izq, *der;
} Nodo;
int selprim(Nodo **pa, int k);
Nodo *ultimos(Nodo *a, int *pk);
```

La función *selprim* debe seleccionar los primeros *k* nodos del árbol en **p*a, modificando el mismo árbol que recibe como parámetro. Además debe preservar la forma del árbol sobreviviente y ***liberar la memoria*** ocupada por los nodos no seleccionados. El nodo *k* de *a* es el nodo *k* al enumarar los nodos de *a en orden*. Esto significa que por ejemplo si el subárbol izquierdo de *a* posee *i* nodos, estos se enumeran recursivamente de 1 a *i* y la raíz de *a* será el nodo *i+1*. Luego se enumeran recursivamente los nodos del subárbol derecho de *a* a partir de *i+2*.

La función *ultimos* retorna un nuevo árbol con los últimos *k* nodos del árbol en *a*. Debe preservar la forma de los últimos *k* nodos. Además debe entregar en **pk* la cantidad efectiva de nodos del árbol retornado.

El siguiente es un ejemplo de uso. El tipo de las variables *r* y *a* es *Nodo** y la de *n* y *k* es *int*. El nodo *r* es el nodo 1, *s* el 2, *t* el 3, *u* el 4, *v* el 5, etc.



Restricciones:

- Por razones de eficiencia, en *selprim* no puede usar *malloc*, ni declarar arreglos o estructuras adicionales.
- En *ultimos* solo debe usar *malloc* para hacer las copias de los últimos *k* nodos. No puede usar otros *malloc* ni declarar arreglos o estructuras adicionales.
- Su solución no puede ser 80% más lenta que la del profesor.
- Se descontará medio punto si su solución no usa la indentación de Kernighan. Además evite que su archivo contenga *tabs*, porque dependiendo del editor usado, puede traducirse en una cantidad distinta de espacios en blanco, arruinando la legibilidad de la indentación.

Instrucciones

Baje *t3.zip* de U-cursos y descomprímalo. El directorio *T3* contiene los archivos (a) *test-seleccion.c* que prueba si su tarea funciona y compara su eficiencia con la solución del profesor, (b) *prof.ref* con el binario ejecutable de la solución eficiente del profesor, (c) *seleccion.h* que incluye los encabezado de las funciones pedidas, y (d) *Makefile* que le servirá para compilar y ejecutar su tarea.

Ejecute el comando *make* sin parámetros bajo Debian 11. Le explicará qué requisitos debe cumplir para aprobar su tarea, cuáles son las opciones de compilación y ejecución, cómo entregar su tarea, cómo borrar los archivos intermedios y cuál es el trabajo del comando *make*. Si no se le ocurre la solución en una hora, lea la ayuda en texto invertido al final del enunciado. ¡Es el algoritmo en palabras! Haga el esfuerzo por entenderlo, porque si no lo entiende, difícilmente se le ocurrirá otra solución. Tradúzcalo a C.

Entrega

Ud. debe entregar por U-cursos el archivo *seleccion.zip* que genera el comando *make zip*. **Recuerde descargar el archivo que subió, descomprimirlo e inspeccionar el contenido para verificar que son los archivos correctos.** Se descuenta medio punto por día de atraso, sin considerar recesos, sábados, domingos y festivos.

Ayuda para selprim: Sea *a*=**pa*. Si *a* es el árbol vacío, retorne 0. Si no, seleccione recursivamente los primeros *k* nodos del subárbol izquierdo. (i) Si logró seleccionar exactamente los *k* nodos, esos son los nodos seleccionados en *a*. Además seleccione recursivamente 0 nodos del subárbol derecho para liberar sus nodos, porque no sobreviven. **Libere la raíz de *a*** porque tampoco sobrevive. (ii) Si solo logró seleccionar *i* nodos, con *i*<*k*, seleccione del subárbol derecho *k-i-1* nodos. El árbol seleccionado tiene la misma raíz del árbol original *a*, el mismo subárbol izquierdo original, y como subárbol derecho los *k-i-1* nodos seleccionados del subárbol derecho.