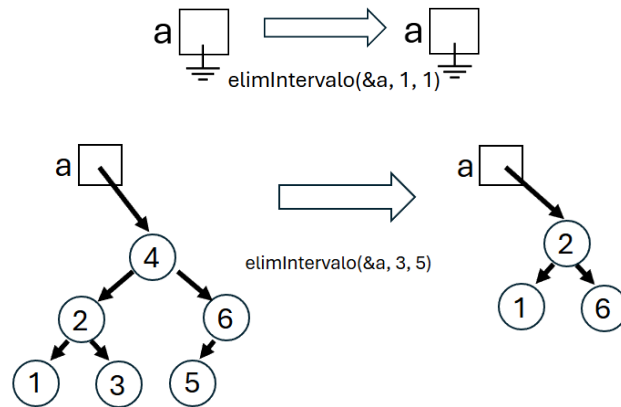


Programa la función:

```
typedef struct nodo {
    int id;
    struct nodo *izq, *der;
} Nodo;
void elimIntervalo(Nodo **pa, int inf, int sup);
```

Esta función recibe en **pa* un árbol de búsqueda binaria (ABB) y entrega en el mismo **pa* un ABB que contiene nodos cuyo *id* no se encuentre en el intervalo **cerrado** [*inf*, *sup*].

Las 2 figuras de más abajo muestran 2 ejemplos de uso. La variable *a* es de tipo *Nodo**.



Restricciones: Debe ser recursivo. No puede pedir memoria adicional con *malloc*. Reutilice los mismos nodos, reasignando los campos *izq* y *der*.

Ayuda: Considere el caso en que el *id* del árbol se encuentra dentro del intervalo [*inf*, *sup*]. En dicho caso puede que haya que eliminar nodos tanto en el subárbol izquierdo como en el derecho, para ello llame recursivamente a la función en ambos subárboles y luego mezcle los dos subárboles resultantes en un nuevo árbol (usando para ello la función **mezclar** que se encuentra en el archivo *intervalo.c*). Finalmente sobrescriba en el puntero *pa* con el nuevo árbol resultante de mezclar y libere el nodo que estaba revisando.

Para el resto de casos razone sobre la relación entre el *id* del nodo que está consultando y el intervalo [*inf*, *sup*] y con ello determinar como seguir con la recursión.

Instrucciones

Baje *t3.zip* de U-cursos y descomprímalo. El directorio *T3* contiene los archivos (a) *test-intervalo.c*, que prueba si su tarea funciona, (b) *intervalo.h* que incluye el encabezado de la función pedida, (c) *Makefile* que le servirá para compilar su tarea, (d) *prof.ref-x86_64* y *prof.ref-aarch64* binarios de benchmark y (e) el archivo *intervalo.c* con función combinar y el cascarón de la función *elimIntervalo* que usted debe programar.

Pruebe su tarea bajo Debian 12 nativo o virtualizado con VirtualBox, Vmware, Qemu o WSL 2. **Ejecute el comando make sin parámetros.** Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo. Su solución no debe ser 80% más lenta que la solución del profesor.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún problema como por ejemplo goteras de memoria.

Cuando pruebe su tarea con *make run* asegúrese que su computador esté en modo alto rendimiento y que no estén corriendo otros procesos intensivo en CPU al mismo tiempo.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *intervalo.zip* generado por el comando *make zip*. **A continuación es muy importante que descargue de U-cursos el mismo archivo que subió, luego descargue nuevamente los archivos adjuntos y vuelva a probar la tarea tal cual como la entregó.** Esto es para evitar que Ud. reciba un 1.0 en su tarea porque entregó los archivos equivocados. Créame, sucede a menudo por ahorrarse esta verificación. Se descontará medio punto por día de atraso. No se consideran los días de receso, sábados, domingos o festivos.