

Programación 3 - Curso 2014

Tarea 2 - Anexo I: Backtracking

1. Objetivos

- Diseñar e implementar una estrategia de Backtracking

2. Conocimientos previos

- Metodología de programación estructurada
- Tipos Abstractos de Datos (TAD)
- Conocimientos básicos de C*
- Multiestructuras
- Backtracking

3. Descripción del problema

La implementación que se solicita es una variante de la realidad planteada en la segunda entrega de la Tarea 2. Asimismo es una extensión de lo ya implementado en la Tarea 1.

3.1. Operaciones

Se agregan nuevas operaciones a los módulos ya implementados en la tarea anterior. Los mismos son:

- `prioridadColaPrioridad` en `ColaPrioridad.h`
- `cantidadPalabraClaveEnArticulo` en `Deposito.h`
- `identificarColecciones` en `OperacionesDeposito.h`

Se recuerda que la resolución de los ejercicios planteados a lo largo de las entregas anteriores debe ayudar y facilitar de forma importante a la implementación de las operaciones solicitadas.

Se les brindará un módulo `Principal.cpp` que funcionará como intérprete de comandos para las operaciones que implementarán. Se recomienda que se utilice archivos de entrada y salida para la ejecución del intérprete. Es decir, se realice de la siguiente forma:

```
$ Principal < entrada.in > salida.out
```

4. Lenguaje a utilizar

El lenguaje a utilizar en este trabajo será C con las siguientes extensiones:

- Operadores `new` y `delete`

- Pasaje de parámetros por referencia (uso de &)
- Declaración de tipos como en C++ para registros y enumerados
- Uso del tipo `bool` predefinido en C++

5. Sobre lo que se espera

Para cada módulo de cabecera `.h` con los prototipos de las operaciones y estructuras solicitadas debe entregarse un módulo `.cpp` con la implementación. Debe respetarse estrictamente los prototipos especificados. Esto es: nombre de la operación, tipo, orden y forma de pasaje de los parámetros y tipo de retorno.

Los módulos de cabecera pueden bajarse de la sección *Letras de las tareas* del sitio en el EVA del curso. Estos módulos no forman parte de la entrega y, por lo tanto, **NO DEBEN SER MODIFICADOS**.

Se brindará un archivo `test.sh`, al igual que en la Tarea 0, de forma que les permita probar la tarea fácilmente.

Los módulos deben funcionar en el ambiente MinGW instalado en facultad. Se espera que todos los módulos compilen sin errores (utilizando las flags `-Wall` y `-Werror`), se ejecuten sin colgarse y den los resultados correctos.

6. Consulta y respuesta de dudas

Las dudas específicas sobre esta parte de la tarea se responderán de forma exclusiva en los foros del EVA del curso. Cuando el cuerpo docente lo considere pertinente se realizarán aclaraciones a través del foro *Novedades*, pero incitamos a los estudiantes estar atentos a lo que se responda en el foro *Consultas sobre Laboratorio*.

7. Fecha de entrega

La fecha de entrega del laboratorio será el **17 de noviembre**. En dicha fecha se deberá entregar, **respetando el formato de los nombres**, los siguientes archivos:

- `ABB_T.cpp`
- `Cola.cpp`
- `ColaPrioridad.cpp`
- `Deposito.cpp`
- `ListaOrd.cpp`
- `OperacionesDeposito.cpp`
- `Pila.cpp`

La primera línea de cada uno de los archivos debe contener un comentario con la cédula de los autores, sin puntos ni dígito de verificación. Por ejemplo, si las cédulas son 1.234.567-8, 4.254.566-2 y 3.339.717-0, la primera línea de cada archivo deberá ser exactamente:

```
/* 1234567 4254566 3339717 */
```

Los archivos deben estar en la raíz del archivo `tarea1.tar.gz` que se entregará según se informará convenientemente. Para generar este archivo sugerimos ejecutar el archivo `test.sh`, en donde se brinda la opción de generarlo. Luego de generar este archivo verifique que en él se encuentran todos los módulos solicitados y que los mismos están en la raíz del mismo.

En caso de no respetarse alguna de las consideraciones anteriores **el trabajo no se considerará aceptable** y deberá ser reentregado si se tratara del primer plazo o no aprobado en caso de ser la reentrega.

8. Sobre la defensa

Aquellas entregas que resulten insuficientes en una primer instancia tendrán la posibilidad de realizar una defensa de su trabajo. Consistirá en reentregar el mismo en un plazo de 24 horas. En dicha reentrega se deberán incluir los cambios que se consideren necesarios para el correcto funcionamiento junto con un archivo denominado `LEAME.txt` con la descripción en lenguaje natural de los cambios hechos. Estos cambios deberán corresponder a una defensa, por lo que **NO** deben alterar la esencia de la entrega inicial.

Para la aprobación de esta defensa la exigencia requerida será mayor que en la primer instancia.

9. Sobre la individualidad del trabajo

Para este obligatorio rige el reglamento de No Individualidad publicado en la página web del curso.