

Tarea 5

Implementación de Tipos Abstractos de Datos

Curso 2018

Índice

1. Introducción	2
2. Materiales	2
3. ¿Qué se pide?	2
3.1. Verificación de la implementación	3
4. Descripción de los módulos y algunas funciones	3
4.1. Módulos de tareas anteriores	3
4.1.1. <i>mayores(cadena_t cad, unsigned int k, unsigned int max)</i>	3
4.2. Módulos nuevos	3
4.2.1. Módulo <i>cola_prioridad</i>	3
4.2.2. Módulo <i>tabla</i>	3
5. Entrega	3
5.1. Plazos de entrega	4
5.2. Identificación de los archivos de las entregas	4

1. Introducción

En esta tarea se continúa la implementación de **Tipos Abstractos de Datos (TADs)**, cumpliendo requerimientos de tiempo de ejecución.

Se agregan los módulos `cola_prioridad` y `tabla`.

Como corresponde al concepto de TAD no se puede usar la representación de un tipo fuera de su propio módulo. Por ejemplo, en `uso_tads.cpp` no se puede usar `rep_cadena` sino las operaciones declaradas en `cadena.h`. Esto será parte de la evaluación.

El correcto uso de la memoria será parte de la evaluación.

El cumplimiento de las restricciones de tiempo será parte de la evaluación.

La tarea otorga un punto a los grupos que la aprueben en la primera instancia de evaluación. La adjudicación de los puntos de las tareas de laboratorio queda condicionada a la respuesta correcta de una pregunta sobre el laboratorio en el segundo parcial. La adjudicación se aplica de manera individual a cada estudiante del grupo.

2. Materiales

Los materiales para realizar esta tarea se encuentran en el archivo *MaterialesTarea5.tar.gz* que se obtiene en la carpeta *Materiales* de la sección *Laboratorio* del sitio EVA del curso ¹.

A continuación se detallan los archivos que se entregan. La estructura de directorios es igual a la de la tarea anterior.

- **info.h**: módulo de definición del tipo `info_t`
- **cadena.h**: módulo de definición de `cadena`.
- **binario.h**: módulo de definición de árbol binario de búsqueda
- **pila.h**: módulo de definición de pilas de elementos de tipo `string`
- **cola_binarios.h**: módulo de definición de colas de árboles binarios de búsqueda
- **iterador.h**: módulo de definición de iteradores sobre elementos de tipo `string`
- **conjunto.h**: módulo de definición de conjuntos de elementos de tipo `info_t`
- **cola_prioridad.h**: módulo de definición de colas de prioridad de elementos de tipo `info_t`
- **tabla.h**: módulo de definición de asociaciones de enteros con `strings`
- **uso_tads.h**: módulo de definición de funciones sobre `tads`
- Casos de prueba en el directorio **test**
- **Makefile**: archivo para usar con el comando `make`, que provee las reglas `principal`, `testing`, `clean` y entrega como en las tareas anteriores.
- **principal.cpp**: módulo principal.

Ninguno de estos archivos deben ser modificados.

¹<https://eva.fing.edu.uy/course/view.php?id=132§ion=6>

3. ¿Qué se pide?

Para cada archivo de encabezamiento **.h**, descrito en la Sección 2, se debe implementar un archivo **.cpp** que implemente **todas** las definiciones de tipo y funciones declaradas en el archivo de encabezamiento correspondiente. Los archivos **cpp** serán los entregables y deben quedar en el directorio **cpp**.

En algunas operaciones se piden requerimientos de tiempo de ejecución. Ese tiempo es el del peor caso, a menos que se especifique otra cosa de manera explícita.

3.1. Verificación de la implementación

Para testear la implementación debe generar el ejecutable y cumplir con todas las pruebas utilizando los archivos **.in** y **.out**. Tenga en cuenta que el archivo **Makefile** proveerá las reglas **principal**, **testing** y **clean** como en las tareas anteriores.

4. Descripción de los módulos y algunas funciones

4.1. Módulos de tareas anteriores

Se mantienen los módulos de la tarea anterior.

En el módulo `uso_tads` se agrega la función `mayores`, de la cual se muestran ejemplos del resultado esperado.

4.1.1. `mayores(cadena_t cad, unsigned int k, unsigned int max)`

cad	k	max	cadena devuelta
<code>[(2,a)]</code>	2	10	<code>[(2,a)]</code>
<code>[(-1,a) (11,a)]</code>	2	10	
<code>[(5,z) (5,a)]</code>	2	10	<code>[(5,z)]</code>
<code>[(5,z) (9,a) (6,g)]</code>	2	10	<code>[(6,g) (9,a)]</code>

4.2. Módulos nuevos

4.2.1. Módulo `cola_prioridad`

En este módulo se implementará una cola de prioridad (de tipo `cola_prioridad_t`) de elementos de tipo `info_t`. La cantidad de elementos que puede contener la cola de prioridad está acotada por un parámetro pasado al crearla. La prioridad de cada elemento está determinada por el valor de un parámetro pasado al insertarlo. Ese valor puede ser modificado luego mediante otras operaciones. El elemento prioritario es el que queda vinculado al valor más pequeño. No puede haber dos elementos vinculados al mismo valor. La cantidad de valores posibles se establece al crear la cola de prioridad.

4.2.2. Módulo `tabla`

En esta módulo se implementa una tabla (de tipo `tabla_t`) de asociaciones de elementos de tipo entero con strings. La cantidad de asociaciones que se pueden mantener en cada momento está acotada por un parámetro pasado al crear la tabla.

5. Entrega

Se mantienen las consideraciones reglamentarias y de procedimiento de las tareas anteriores.

Se debe entregar el siguiente archivo, que contiene los módulos implementados `binario.cpp`, `cadena.cpp`, `cola_binarios.cpp`, `cola_prioridad.cpp`, `conjunto.cpp`, `info.cpp`, `iterador.cpp`, `pila.cpp`, `tabla.cpp`, `uso_tads.cpp`:

- **Entrega5.tar.gz**

Este archivo se obtiene al ejecutar la regla `entrega` del archivo `Makefile`:

```
$ make entrega
tar zcvf Entrega5.tar.gz -C src info.cpp cadena.cpp binario.cpp pila.cpp cola_binarios.cpp conjunto
info.cpp
cadena.cpp
binario.cpp
pila.cpp
cola_binarios.cpp
conjunto.cpp
iterador.cpp
cola_prioridad.cpp
tabla.cpp
uso_tads.cpp
```

NO SE PUEDEN ENTREGAR MÓDULOS ADICIONALES A LOS SOLICITADOS

5.1. Plazos de entrega

El plazo para la entrega es el **miércoles 20 de junio a las 14 horas**.

NO SE ACEPTARÁN ENTREGAS DE TRABAJOS FUERA DE FECHA Y HORA. LA NO ENTREGA O LA ENTREGA FUERA DE LOS PLAZOS INDICADOS IMPLICA LA PÉRDIDA DEL CURSO.

5.2. Identificación de los archivos de las entregas

Cada uno de los archivos a entrega debe contener, en la primera línea del archivo, un comentario con el número de cédula de el o los estudiantes, **sin el guión y sin dígito de verificación**. Ejemplo:

```
/* 1234567 - 2345678 */
```