



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Estructuras de Datos, 2020-10
Parcial 1 - 4 de marzo de 2020

Reglas y recomendaciones

Durante el parcial, deben observarse las siguientes reglas:

1. El parcial se debe desarrollar en el computador, escribiendo las respuestas en un archivo de texto plano (TXT). Guarde regularmente su trabajo (para evitar posibles pérdidas de información), con el nombre de archivo
`par1_apellido1_apellido2_nombre1_nombre2.txt`
Tenga en cuenta que los nombres y apellidos van todos en minúsculas, los espacios se reemplazan por guiones bajos y no se utilizan ni tildes ni 'ñ'. El archivo de respuestas debe enviarse únicamente a través de la actividad correspondiente en UVirtual.
2. Archivos adicionales se aceptarán sólo en el caso del diseño, de acuerdo a las instrucciones que se encuentran en el enunciado del parcial. El archivo debe guardarse como
`par1_apellido1_apellido2_nombre1_nombre2_dis.pdf`
y enviarse junto con el archivo de respuestas a través de la actividad correspondiente en UVirtual.
3. El parcial tiene una duración de dos horas, contadas a partir del inicio normal de la clase. La actividad en UVirtual está programada para cerrarse al terminar las dos horas, por lo que cualquier archivo no entregado dentro de ese lapso se considerará como parcial no presentado y tendrá una calificación de 0.0.
4. El parcial es estrictamente individual y se debe desarrollar únicamente en la sala de computadores del día.
5. Se recomienda no utilizar compiladores, intérpretes de lenguajes de programación o entornos de desarrollo de cualquier tipo.
6. Puede utilizar sus apuntes, libros, e Internet para obtener la información que necesite para el parcial. Sin embargo, está absolutamente prohibido comunicarse con cualquier otro ser humano para obtener información sobre el parcial, a través de cualquier medio (conversación directa, MSN, Skype, gtalk, whatsapp, blackberry messenger, etcétera).
7. Los celulares deben permanecer apagados, y no se debe enviar ni recibir ningún mensaje de texto.
8. La única excepción a lo anterior son los profesores y monitores de la asignatura, quienes sólo responderán consultas respecto a la claridad de las preguntas del parcial y no responderán consultas sobre la materia.
9. Si el estudiante incumple con cualquiera de las reglas, será evaluado con nota 0.0
10. **Recuerde que diseñar no implica implementar.**

1 (16%) Análisis de código

Considere la siguiente función (se garantiza que los argumentos contienen la cantidad de elementos necesarios):

```
typedef std::vector< unsigned int >    TVector;
typedef std::list< long >              TList;
typedef std::list< TList >             TList2;
void func( TList2& l, TVector& v )
{
    v.clear( );
    TList2::iterator l2It = l.begin( );
    for( ; l2It != l.end( ); l2It++ )
    {

        TList::iterator l1It = l2It->begin( );
        unsigned int i = 0;
        for( ; l1It != l2It->end(); l1It++ )
        {
            if ( *l1It / 3 == 0 )
            {
                i++;
            }
        }
        v.push_back( i );
    }
}
```

1.1 ¿Qué hace «func»? (describala en 12 palabras o menos)

1.2 ¿Cuál es el orden de complejidad de «func»? Justifique brevemente su respuesta.

2 (18%) Selección múltiple con única respuesta

2.1 Se tienen 3 algoritmos para ordenar un arreglo de datos, con los siguientes tiempos de ejecución: **algoritmo 1**, $T(n) = 15n + 2$; **algoritmo 2**, $T(n) = 7n + 2n^2 + 3$; y **algoritmo 3**, $T(n) = 4 + 11 \log n$. ¿Cuál es el algoritmo de mayor complejidad teórica?

1. algoritmo 2 con $O(2^n)$.
2. algoritmo 2 con $O(n^2)$.
3. algoritmo 1 con $O(n)$.
4. algoritmo 3 con $O(\log n)$.
5. algoritmo 3 con $O(n)$.

2.2 En una lista doblemente encadenada de la STL (`std::list< T >`) las siguientes operaciones tienen orden de complejidad $O(n)$:

1. inserción o eliminación de un elemento en el inicio.
2. inserción o eliminación de un elemento en el final.
3. inserción o eliminación de un elemento en una posición intermedia.
4. ninguna de las anteriores.
5. todas las anteriores.

3 (66%) Diseño e Implementación de TADs

La empresa *Tralala* se encarga de producir y comercializar al menos 9 diferentes variedades de dulces y golosinas. El proceso de producción está claro y bien establecido, sin embargo, las directivas de la empresa esperan evaluar el proceso de empaçado de los dulces en busca de oportunidades de mejora. Cada dulce está identificado por un nombre, un sabor, y un número identificador único de varios dígitos, donde el primer dígito corresponde al tipo o variedad de dulce. Al empacarlos, cada paquete de dulces toma el nombre y el sabor de los dulces que almacena, pero además incluye la cantidad de dulces que ese paquete contiene.

El empaçado de los dulces tiene 3 fases bien diferenciadas. En la primera fase los dulces de todas las variedades son apilados en una estructura (contenedor A) con capacidad para 50 dulces. Una vez se ha llenado esta estructura, se pasa a la fase dos, donde una operadora retira manualmente uno a uno los dulces del contenedor A y los va ubicando cada uno en la banda transportadora que le corresponde, ya que hay una banda por cada tipo de dulce producido (identificada por un número dígito del 1 al 9). Cada banda transportadora también tiene una capacidad límite, que corresponde a la capacidad del paquete en el que se van a empacar los dulces. La repartición de dulces en las bandas sólo se hace cada vez que se llena completamente el contenedor A. Cuando cada banda transportadora se llena, en la fase tres otra operadora ubica todos los dulces de esa banda en un paquete, y almacena este paquete en un contenedor B, del cual se tomarán posteriormente para la distribución.

La empresa *Tralala* considera implementar estos procesos en un sistema de información, para lo cual han identificado que en la fase 1 de empaçado se cuenta con un único contenedor A tipo LIFO de capacidad 50 dulces, en la fase 2 se cuenta con hasta 9 bandas transportadoras tipo FIFO, con capacidad variable de dulces para cada una, y en la fase 3 se cuenta con un único contenedor B secuencial que almacena los paquetes de dulces. En el proceso de empaçado, se han identificado inicialmente dos procedimientos importantes, que pueden ser desarrollados en el sistema de información:

1. Repartir los dulces del contenedor A, es decir, realizar el proceso descrito en la fase dos, donde se verifica que el contenedor esté lleno y luego se procede a ubicar los dulces en sus correspondientes bandas transportadoras, para que después puedan ser empaquetados.
2. Empaquetar los dulces de las bandas transportadoras, es decir, realizar el proceso descrito en la fase tres, donde se verifica que cada banda transportadora esté llena a su capacidad máxima, y luego se retiran los dulces en ella para ubicarlos en un paquete que se almacena en el contenedor B.

Se le pide entonces diseñar e implementar (en C++) los componentes ya descritos del sistema de información para la empresa *Tralala*.

3.1 (20%) Diseño

Diseñe el sistema y el (los) TAD(s) solicitado(s). Recuerde que diseñar es un proceso previo a la implementación, por lo que no debería contener ninguna referencia a lenguajes de programación (es **decir, si escribe encabezados o código fuente, el punto no será evaluado y tendrá una calificación de cero**). Para simplicidad del diseño, no es necesario incluir los métodos obtener y fijar (*get / set*) del estado de cada TAD. Para el diagrama de relación entre TADs, anéxelo en formato PDF como parte de su entrega.

Algoritmos

De acuerdo al diseño realizado en el punto anterior, escriba la implementación de los siguientes algoritmos en C++. Estas implementaciones deberán tener en cuenta:

- la definición apropiada de los prototipos de los métodos/funciones (i.e. recibir/retornar los datos suficientes y necesarios para su correcta ejecución),

- el correcto uso del (de los) TAD(s) diseñado(s) en el punto anterior,
- la escritura de todo el código que pueda llegar a necesitar que no esté incluido en la STL (excepto los métodos *get* / *set* del estado de cada TAD), y
- el **NO** uso de salidas/entradas por pantalla/teclado (i.e. paso/retorno correcto de valores y/o objetos, **si usa “cout”, “printf”, “cin”, “scanf”, “cerr” o cualquier otro mecanismo de entrada/salida por flujos el punto no será evaluado y tendrá una nota de cero**),

3.2 (22%) Repartir los dulces del contenedor A.

3.3 (24%) Empaquetar los dulces de las bandas transportadoras.