
Universidad Tecnológica Metropolitana.

Departamento de Computación e Informática.

Análisis de Algoritmos.

Profesor: Sebastián Salazar Molina.

Proyecto

07 de junio del 2020

VISIÓN GENERAL

La capacidad de ordenar y clasificar información es una necesidad básica en cualquier sistema de información, encontrar y usar algoritmos eficientes, tiene un enorme impacto en el desempeño de los sistemas.

OBJETIVOS

1. Investigar y usar algoritmos de ordenamiento.
2. Investigar y usar algoritmos de búsqueda.
3. Mejorar el uso de archivos.

ESPECIFICACIONES

La Universidad Tecnológica Metropolitana posee un conjunto de carreras, que tienen una serie de restricciones de ingreso. Por otro lado, los estudiantes postulan a la universidad rindiendo la Prueba de Selección Universitaria que junto al promedio de Notas de Enseñanza Media y al lugar que obtuvieron en la promoción de su cohorte en su etapa de educación secundaria, entregan un puntaje medible para postular a diversas carrera.

Se solicita desarrollar una aplicación C/C++, que utilizando un archivo de entrada (que contiene los puntajes obtenidos por el estudiante). debe entregar como salida un conjuntos de archivos txt (de texto plano), en donde cada archivo debe corresponder a una carrera. Los estudiantes (representados por su rut y el puntaje de postulación, ordenados de mayor a menor) sólo pueden clasificarse en una carrera a la vez y como máximo sólo pueden cumplirse los cupos disponibles. Los estudiantes que no alcancen un cupo deben ignorarse.

Equipo.

El proyecto deberá ser realizado en forma grupal, con los equipos conformados en clases.

Informe.

Se debe presentar un informe con la siguiente estructura:

- Debe tener una introducción.
- Debe indicar la forma de resolver el problema.
- Debe entregar la complejidad de los algoritmos usados y métricas de rendimiento.
- No debe superar las 15 páginas.

Código.

El código debe ser entregado a más tardar, el 07/06/2020 hasta las 23:59:59.999 horas de Santiago de Chile, este proyecto debe estar respaldado en un repositorio personal github.

EVALUACIÓN

Documentación.

Parte de la evaluación consiste en la documentación de las funciones. Que debe ser clara, concisa y descriptiva de lo que el código realiza.

Código

El código debe ser claro, fácil de leer, ordenado y cumplir con buenas prácticas de programación, se inspeccionará el código fuente.

Resultados.

Un criterio de evaluación que se tomará en consideración: el tiempo de ejecución de la tarea. Menos es mejor. La evaluación es porcentual.

Aplicación

Implementación.

La implementación debe realizarse en C/C++, usando el compilador GCC (versión 9.3) incluido en Ubuntu 20.04 LTS. Se valora el uso de OpenMP.

Operaciones

La aplicación debe ejecutar las siguientes operaciones.

Generar Archivos.

Ejemplo:

`./dist/programa 1 /media/Compartido/puntajes.csv /media/Compartido/Evaluacion02/Grupo`

- Clasificar los estudiantes en función de su mejor opción de postulación.
 - Este archivo tiene la estructura:
 - Rut.
 - NEM.
 - Ranking.
 - Matemática.
 - Lenguaje.
 - Ciencias.
 - Historia.
 - Datos de salida exitoso.
 - Archivo txt cuyo nombre será el código de la carrera.txt (Ejemplo: 21041.txt), almacenado en la carpeta definida por la línea de comandos..
 - Este archivo tiene la estructura.
 - Rut
 - Puntaje ponderado para acceder a la carrera.
 - Datos de salida fallido
 - Mensaje descriptivo del error.

Buscar rut.

<code>./dist/programa 2 14123123 /media/Compartido/Evaluacion02/Grupo</code>
--

- Busca el rut ingresado en los archivos existentes en la carpeta especificada.
- Si el rut es encontrado, debe mostrar en pantalla, el nombre del archivo en el cual se encontró y la línea dónde el registro se encuentra.
- Si el rut no es encontrado, se debe mostrar un mensaje descriptivo.

Datos de entrada

El programa debe leer tres argumentos, por la línea de comandos:

1. Código numérico.
 - a. Si el código es **1**. El programa generará los archivos con las postulaciones.
 - b. Si el código es **2**. El programa hará una búsqueda de un rut sobre los archivos generados.
2. Texto
 - a. Si el código anterior es **1**, este argumento corresponderá al archivo de puntajes (puntajes.csv).
 - b. Si el código anterior es **2**, este argumento corresponderá al rut sobre el cuál se deberá realizar la búsqueda.
3. El tercer argumento corresponde a la carpeta donde se almacenarán los archivos de salida.

En caso de no ingresar datos, se debe mostrar la información de los integrantes del grupo.

Datos de salida.

- Si la opción es 1, debe generar los archivos.
- Si la opción es 2, debe mostrar la información de la búsqueda.
- Si no hay opciones, debe mostrar la información de los integrantes del grupo.