

Tarea 3

Programación orientada al objeto

1^{er}. Semestre 2018

Introducción

Esta tarea está diseñada para que Ud. experimente con la creación de plantillas genéricas del lenguaje de programación C++.

1. El método de búsqueda de la proporción áurea

El método de búsqueda de la *proporción áurea* es un método de búsqueda similar a la búsqueda binaria que, en vez de dividir el universo de búsqueda en dos secciones iguales, lo hace en secciones desiguales guiadas por la proporción áurea.

La estrategia de búsqueda de este método se basa en tener tres puntos iniciales: los extremos del intervalo, (x_1, x_2) y un tercer punto x_3 entre los dos primeros tal que la relación entre la distancia de este punto interno al extremo x_2 , $x_2 - x_3$, y la distancia entre los extremos, $x_2 - x_1$, es siempre una constante:

$$\frac{x_2 - x_3}{x_2 - x_1} = \frac{\sqrt{5} - 1}{2} = \tau = 0.618034 \dots$$

Entonces, el punto x_3 divide al intervalo (x_1, x_2) en dos subintervalos, donde el intervalo (x_1, x_3) es el 61.8 % del intervalo (x_3, x_2) . El punto a buscar está dentro de uno de los dos intervalos, y el método itera generando el siguiente punto en el intervalo donde se encuentra el dato a buscar.

El algoritmo opera, entonces, como sigue:

1. Para un vector de n datos, se hace v_0 el punto x_1 y v_{n-1} el punto x_2 , y se comprueba si alguno de ellos es el dato buscado. En caso afirmativo, se termina la búsqueda.
2. Se calcula la posición de x_3 de acuerdo al método antes descrito, y se comprueba si x_3 es el dato buscado.
3. Si x_3 es mayor al dato buscado, se hace $x_2 = x_3$ y se repite la búsqueda.
4. Si x_3 es menor al dato buscado, se hace $x_1 = x_3$ y se repite la búsqueda.
5. Si x_3 es igual al dato buscado, se termina la búsqueda.

Por supuesto, si en algún momento $x_1 == x_2$ y ambos son distintos al dato buscado, entonces éste no se encuentra en el vector.

Entonces, se le solicita escribir una función genérica en C++ utilizando plantillas llamada `bool GoldenSectionSearch(vector<T>& vec, const T& dato)` que busque un elemento de valor `dato` de tipo `T` en el vector `vec` utilizando el algoritmo de búsqueda de la sección áurea antes expuesto. Su función debe operar correctamente sobre diferentes tipos de datos como `int`, `double`, `string`, etc. Recuerde que los datos deben estar ordenados para que este algoritmo funcione.

Adjunto a este listado, Ud. encontrará el archivo `MillonDeInts.txt`, que contiene 10000 números enteros y el archivo `MillonDeDoubles.txt`, que contiene 1000000 números de punto flotante de doble precisión. Su código debe imprimir en pantalla el número de comparaciones realizadas para cada búsqueda. Puede probar entonces su código buscando los valores 0 y 0.0, que no están presentes en los archivos respectivos.

Sugerencia: desarrolle primero su código para un vector de enteros pequeño, luego verifique su funcionamiento para el archivo dado, y finalmente modifíquelo para que opere sobre diferentes tipos de datos. Tenga especial cuidado con las condiciones de borde en el límite de la representación.

2. Buscando anagramas

Ud. ha sido contratado por una agencia gubernamental de esas que no existen y donde todos trabajan vestidos de negro para ayudar en la búsqueda de terroristas internacionales pertenecientes a una misteriosa organización llamada *La Organización Misteriosa*, ó LOM. Se sabe que esta organización tiene agentes de todas partes del mundo diseminados por todo el mundo, y que es muy difícil identificarlos. Sin embargo, esta organización tiene una debilidad por los anagramas: todos sus agentes usan alias generados como anagramas de su nombre y apellido. Por ejemplo, el peligroso terrorista **Andres Cheke**, capturado el año pasado, usaba como chapa el nombre **Derek Chesna**, un anagrama de su nombre. El anagrama debe ser completo: no deben faltar ni sobrar letras.

Se le pide, entonces, desarrollar una aplicación en C++ que reciba como entradas el nombre, apellido y sexo del posible agente de *LOM*, y que muestre en pantalla una lista de todos los pares (nombre, apellido) posibles para el alias. Para ello, Ud. dispone de tres archivos de datos, conteniendo los apellidos, nombres femeninos y nombres masculinos más comunes en Estados Unidos.

Su programa debe usar iteradores y al menos tres llamadas a funciones genéricas diferentes de la biblioteca STL. Comente su código para indicar su proceso de solución.

El siguiente esqueleto de código le puede servir como punto de partida para su solución. En éste se muestra cómo leer los archivos a vectores de `string`. Ud. es libre de reemplazarlos por otros contenedores, si así lo desea.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

using namespace std;

int main() {
    // Lee archivos de datos
```

```

ifstream apellidos_US("Apellidos_US.txt");
ifstream hombres_US("NombresHombres_US.txt");
ifstream mujeres_US("NombresMujeres_US.txt");

string temp;
vector<string> apellidos;
vector<string> hombres;
vector<string> mujeres;

while(apellidos_US >> temp) {
    apellidos.push_back(temp);
}
while(hombres_US >> temp) {
    hombres.push_back(temp);
}
while(mujeres_US >> temp) {
    mujeres.push_back(temp);
}

string nombre, apellido, sexo;
cout << "Ingrese el nombre a buscar: ";
cin >> nombre;
cout << "Ingrese el apellido a buscar: ";
cin >> apellido;
cout << "Ingrese el sexo (M/F): ";
cin >> sexo;

// INSERTE SU CODIGO AQUI

return 0;
}

```

Esta tarea puede ser realizada en grupos de máximo 3 personas. Su código fuente deberá ser subido a InfoAlumno antes de las 6 pm del día 22 de junio. Además, deberá entregar un informe escrito en secretaría de Electrónica antes de esa fecha, que *debe* incluir un listado de su programa, además de una descripción de las dificultades encontradas y cómo fueron solucionadas.