Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2022.

PROYECTO1: ANALIZADOR DE PISOS

202012039 – Angel Francisco Sique Santos

Resumen

Se solicitó crear un robot capaz de comparar patrones de pisos y decidir la manera más barata de cambiar de un patrón a otro, además de ordenarlos e imprimir el resultado haciendo uso de la herramienta Graphviz.

Para resolver el problema se implementó el uso de datos abstractos como listas simples simulando una matriz para comparar cada casilla con una segunda matriz para comparar sus atributos y los de sus casillas alrededor y decidir qué movimiento es el adecuado para hacer.

Palabras clave

Listas

Graphviz

Matriz

Programación orientada a objetos

Python

Abstract

It was requested to create a robot capable of comparing floor patterns and deciding the cheapest way to change from one pattern to another, in addition to ordering them and printing the result using the Graphviz tool.

To solve the problem, the use of abstract data such as simple lists was implemented, simulating a matrix to compare each box with a second matrix to compare its attributes and those of its surrounding boxes and decide which move is the right one to make.

Keywords

Lists

Graphviz

Matrix

Object-oriented programming

Python

Introducción

El tema principal de este proyecto es las estructuras de datos. La manera de implementación en este caso es de listas simplemente enlazadas, el cómo recorrerlas, buscar entre ellas y como ordenarlas fue lo que se necesito para cubrir las necesidades que se nos plantean en el problema de este proyecto.

Para el ordenar en orden alfabético se usó el método de la burbuja.

Para mostrar gráficamente el patrón inicial y final se utilizó la herramienta Graphviz.

El tipo de archivos permitido para la lectura es XML, para leerlo se implementó ElementTree.

Desarrollo del tema

Los tipos de datos abstractos utilizados para resolver el problema fueron:

a. Listas enlazadas simples:

Una lista enlazada simple es una estructura de datos en la que cada elemento apunta al siguiente. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma. La figura 1 representa esta estructura de datos.

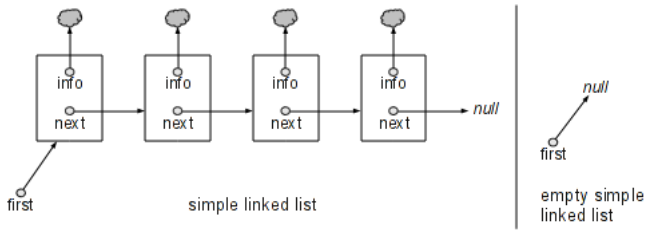

Figura 1. Lista enlazada.

Fuente: http://www.it.uc3m.es/java/2011-12/units/pilas-colas/guides/2/guide_es_solution.html.

La implementación en el proyecto fue para todas las listas donde se almacenaban los datos al leerlos del XML.

b. Método de la burbuja:

El ordenamiento de burbuja (Bubble Sort en inglés) es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran pequeñas "burbujas". También es conocido como el método del intercambio directo. Dado que solo usa comparaciones para operar elementos, se lo considera un algoritmo de comparación, siendo uno de los más sencillos de implementar.

Este algoritmo realiza el ordenamiento o reordenamiento de una lista a de n valores, en este caso de n términos numerados del 0 al n-1; consta de dos bucles anidados, uno con el índice i, que da un tamaño menor al recorrido de la burbuja en sentido inverso de 2 a n, y un segundo bucle con el índice j, con un recorrido desde 0 hasta n-i, para cada iteración del primer bucle, que indica el lugar de la burbuja.

La burbuja son dos términos de la lista seguidos, j y j+1, que se comparan: si el primero es mayor que el segundo sus valores se intercambian.

Esta comparación se repite en el centro de los dos bucles, dando lugar a una lista ordenada. Puede verse que el número de repeticiones solo depende de n y no del orden de los términos, esto es, si pasamos al algoritmo una lista ya ordenada, realizará todas las comparaciones exactamente igual que para una lista no ordenada. Esta es una característica de este algoritmo.

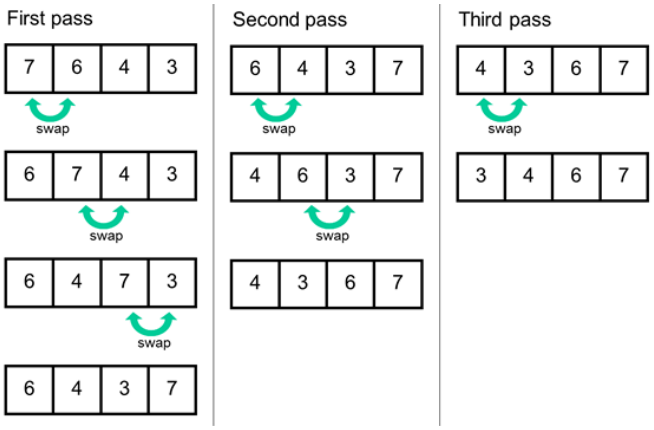

Figura 2. Método burbuja.

Fuente: https://medium.com/austins-software-engineering-journey/bubble-sort-b2f0d63e38f7.

c. Graphviz.

Graphviz es un programa de visualización gráfica de fuente abierta. La visualización de gráficos es una forma de representar información estructural como diagramas de redes y gráficos abstractos. Tiene aplicaciones importantes en redes, bioinformática, ingeniería de software, diseño web y de bases de datos, aprendizaje automático y en interfaces visuales para otros dominios técnicos.

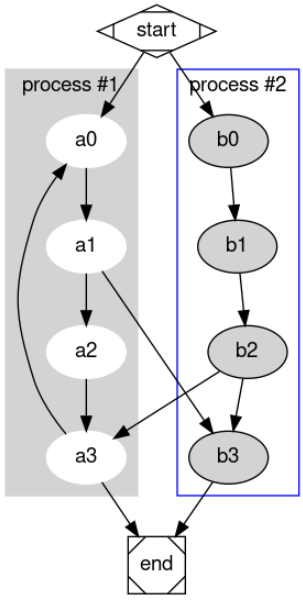

Figura 3. Ejemplo Graphviz.

Fuente: https://graphviz.org/.

d. ElementTree

El módulo xml.etree.ElementTree implementa una API simple y eficiente para parsear y crear datos XML.

d1. XML

XML es un formato de datos inherentemente jerárquico, y la forma más natural de representarlo es con un árbol. ET tiene dos clases para este propósito - ElementTree representa todo el documento XML como un árbol, y Element representa un solo nodo en este árbol. Las interacciones con todo el documento (leer y escribir en/desde archivos) se realizan normalmente en el nivel de ElementTree. Las interacciones con un solo elemento XML y sus sub-elementos se realizan en el nivel Element.

```
<?xml version="1.0"?>
<pisosArtesanales>
    <piso nombre="ejemplo01">
    <R> 2 </R>
    <C> 4 </C>
    <F> 1 </F>
    <S> 1 </S>
    <patrones>
        <patron codigo="zcod11">WBWBWWWB</patron>
        <patron codigo="dcod12">BWBWWWWW</patron>
        <patron codigo="acod13">WWWBBBBW</patron>
        <patron codigo="hcod14">BWBWBWBW</patron>
        <patron codigo="ecod15">WBBWWBBW</patron>
        <patron codigo="vcod16">BWWBBWBW</patron>
    </patrones>
    </piso>
    <piso nombre="ejemplo02">
    <R> 3 </R>
    <C> 3 </C>
```

Figura 4. Ejemplo del XML.

Fuente: Elaboración propia

Conclusiones

Las listas enlazadas simples fueron lo más indispensable en este proyecto ya que fueron la base de todo, no se pudiera implementar de otra manera tan eficiente todo lo requerido y al usar listas simples se optimiza también el uso de espacio en memoria que si se usara otro tipo de lista que también podría funcionar, pero malgasta más memoria.

El uso de Graphviz refuerza conocimientos sobre enlazar nodos unos con otros, ya que es para esto que se utiliza esta herramienta y además que esto es indispensable en la programación orientada a objetos donde siempre se está enlazando objetos con otros.

El uso de Python para resolver este problema fue agradable ya que su modalidad de uso es un poco más simplificada que lenguajes como Java, aunque se podría resolver el mismo problema con otro lenguaje, se siente más simplificado trabajar con Python. Es buena forma de reforzar habilidades con este lenguaje y mejorar la lógica al resolver este tipo de problemas.

Referencias bibliográficas

Anónimo. (2019). *Listas simples enlazadas*. Obtenido de uc3m: http://www.it.uc3m.es/java/2011-12/units/pilas-colas/guides/2/guide_es_solution.html

graphviz.org. (10 de Agosto de 2021). *¿Qué es Graphviz?* Obtenido de Graphviz: https://graphviz.org/

python.org. (6 de Marzo de 2022). *ElementTree*. Obtenido de python.org: https://docs.python.org/es/3/library/xml.etree.elementtree.html

Wikipedia. (2 de Marzo de 2022). *Método burbuja*. Obtenido de wikipedia: https://es.wikipedia.org/wiki/Ordenamiento_de_burbuja

Apéndices

a. Diagrama de clases: