DOCUMENTO TÉCNICO HASH V DISYUNTOS

[https://github.com](https://github.com/Nat15005/Hash_Disyuntos)

Especificación:

Se solita implementar las diferentes versiones de tabla hash, donde no se utilice una estructura de datos no indexada como colección en profundidad, es posible implementar bien sea una linked list, un árbol BST o una cola de prioridad. Por otra parte, es solicitado extender el prototipo funcional presentado en el laboratorio anterior pero esta vez añadiendo un reporte/característica usando Tablas de hash y la implementación vista de conjuntos disyuntos.

Adjuntar casos de prueba en el informe.

Entrada:

Pedido

Salida:

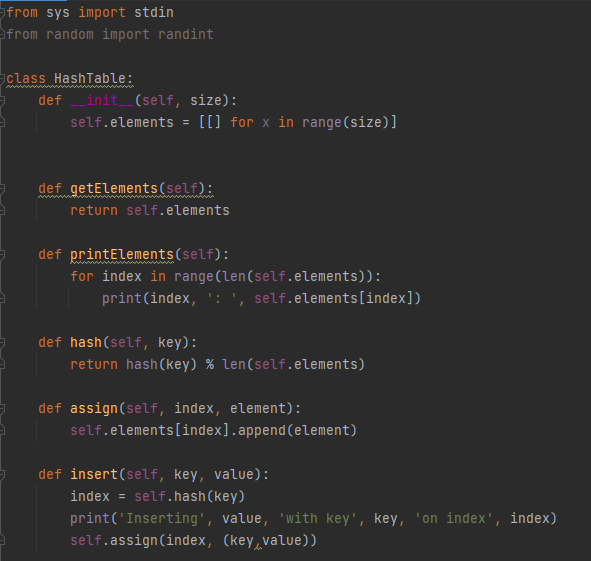
Resumen del pedido

Diseño:

Estrategia

Quisimos plantear un escenario en el cual se conectarán muchas relaciones de forma no dirigida, con el objetivo de ver maneras óptimas de llevar pedidos de un lugar a otro, por ende, escogimos una empresa de mensajería, escogimos dos ciudades: Bogotá y Medellín, ahora se va a plantear una implementación con hash.

Algoritmo y Documentación



Texto

Descripción generada automáticamente

Casos de Prueba

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Justificación** | **Salida** |
| 0  1  Lara 14/02/2023  0 | Probar formato de archivo de pedido | [('Lara', '14/02/2023')] |
| 0  1  Lara 14/02/2023  Hora  Lara  0 | Probar búsqueda de pedido por hora a partir del nombre | [('Lara', '14/02/2023')]  14/02/2023 |
| 1  Lara 14/02/2023  Hora  Lara  resumen  0 | Probar resumen de los pedidos | [('Lara', '14/02/2023')]  14/02/2023  0 : [('Lara', '14/02/2023')] |

Fuentes

HashLab13.py

**CONJUNTOS DISYUNTOS**

Entrada:

Para crear un conjunto que represente más de un pedido se debe ingresar como “nombre1, nombre2”, el resto de acciones son intuitivas y se especifican.

Salida:

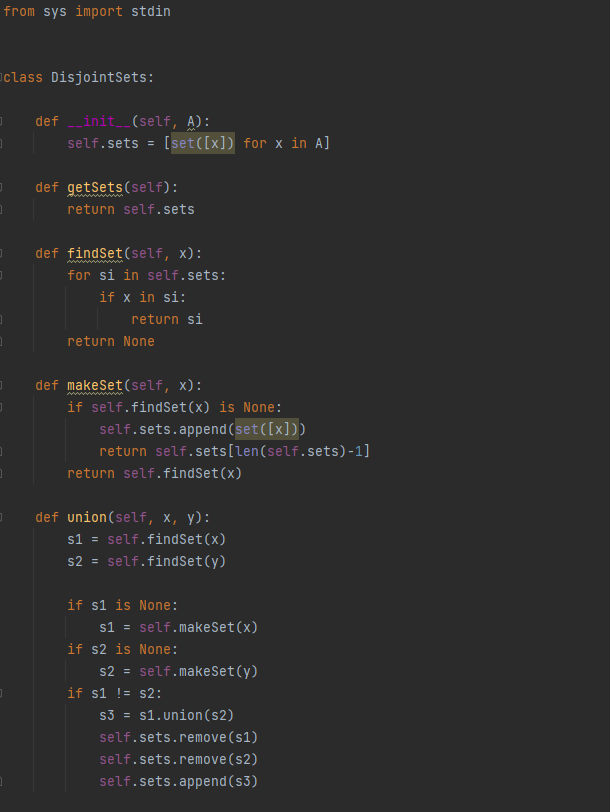
Resumen del pedido

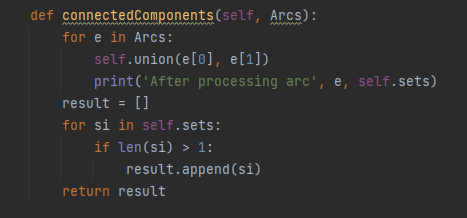
Diseño:

Estrategia

Primero se inicia el programa con la entrada especificada, luego se ingresa en una sola línea todos los juntos del conjunto disyunto, si un conjunto tiene mas de 1 elemento se especifica como se debe ingresar. La salida es el resultado del conjunto disyunto, después de su creación o en su defecto después de realizar cualquiera de las especificaciones ofrecidas.

Algoritmo y Documentación





Casos de Prueba

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Justificación** | **Salida** |
| 0  anuel,barbara camilo andres,juan fran,fred esteban  1  pepe  2  camilo esteban  3  0 | Probar pedidos cuando se ingresan 1 y 2 ordenes de clientes | Sus pedidos son: [{'anuel,barbara'}, {'camilo'}, {'andres,juan'}, {'fran,fred'}, {'esteban'}] |
| 0  anuel,barbara camilo andres,juan fran,fred esteban  1  pepe  2  0 | Probar pedidos agregando una orden | Sus pedidos son: [{'anuel,barbara'}, {'andres,juan'}, {'fran,fred'}, {'pepe'}, {'camilo', 'esteban'}] |

Fuentes

DisjointSets13.py