

9 nulye

Estructura de Datos

Universidad Nacional de Tres de Febrero

Nombre y apellido: Lucas Nevado

Primer Parcial

20/09/2016

1. Un consultorio tiene un archivo CSV de citas de pacientes con campos "Fecha y hora", "Nombre", "Obra Social", "Código de paciente", "Tipo de Consulta" ordenado por Fecha y hora. Como resultado, se debe generar un archivo de texto que provea información de detalle para cada Obra Social. El detalle debe ser: para cada Obra Social, todos los pacientes que se atendieron ordenados por fecha y hora, en qué fecha y hora y cuál es el tipo de consulta. Además se debe proveer un resumen de total de pacientes para cada obra social. El orden en el que se muestran las obras sociales no es importante.

Por ejemplo:

Si el archivo CSV contiene

"Fecha y hora", "Nombre", "Obra Social", "Código de paciente", "Tipo de Consulta"

"2015-10-09 09:30:00", "Juan", "OSECAC", "30-1", "1"

"2015-10-09 10:00:00", "Ana", "OSDE", "60-2", "1"

"2015-10-09 10:30:00", "Jorge", "OSDE", "60-3", "2"

"2015-10-09 11:30:00", "Luz", "OSECAC", "30-4", "2"

la salida debe ser

OSECAC:

"2015-10-09 09:30:00", "Juan", "30-1", "1"

"2015-10-09 11:30:00", "Luz", "30-4", "2"

OSECAC. Total 2 pacientes.

OSDE:

"2015-10-09 10:00:00", "Ana", "60-2", "1"

"2015-10-09 10:30:00", "Jorge", "60-3", "2"

OSDE. Total 2 pacientes.

2. Se tiene un diccionario de recetas de comida con clave el nombre del plato y como contenido una lista de pares (ingrediente, cantidad) con cantidades para una porción. Se tiene otro diccionario de productos en mi alacena con clave el ingrediente y como contenido la cantidad disponible. (a) Dado una lista que_se_cocina que contiene pares (plato, cantidad de porciones) construir un diccionario lista de compras, con los ingredientes que faltan para cocinar lo planificado para hoy. Por ejemplo: si

recetas = {'pizza': [(('harina', 30), ('tomate':1), ('queso': 25)], 'ensalada': [(('lechuga', 100), ('tomate':1), ('zanahoria': 1)], 'flan': [(('huevo':1), ('leche':100), ('azucar':60))]}

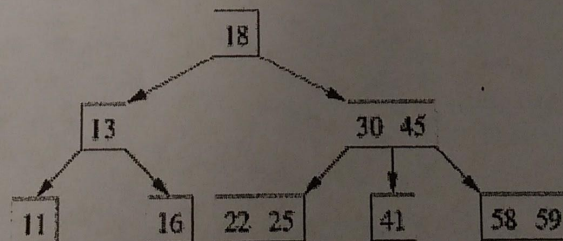
alacena = {'huevo': 6, 'harina':500, 'queso': 250, 'tomate':2}

que_se_cocina = [(('pizza',4), ('ensalada', 2)]

la lista de compras debe ser: {'tomate': 5, 'lechuga': 200, 'zanahoria': 2 }.

(b) Escribir funciones Python para hacer persistentes estos diccionarios y para recuperarlos.

3. Dado el árbol B de orden 3 de la figura, insertar los valores 23, 55, 53, en ese orden, paso a paso.

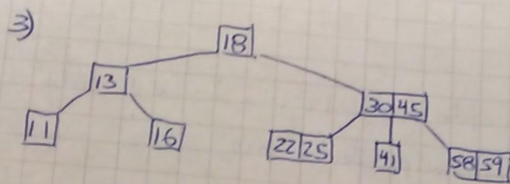


Lucas Nevado 40768

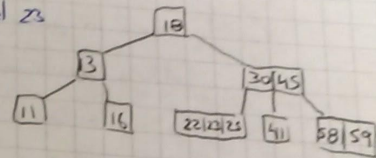
Estructura de Datos

1/2

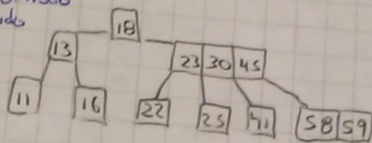
20/9/16



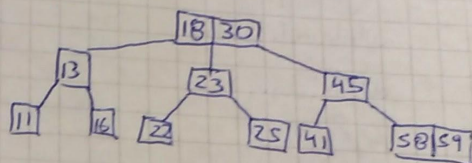
Inserto el 23



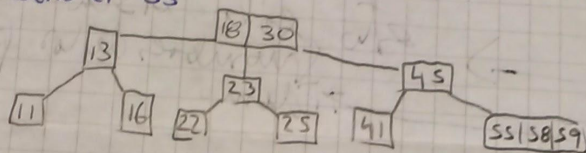
Reacomodo
y Divide



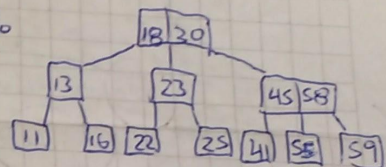
Reacomodo
y Divide



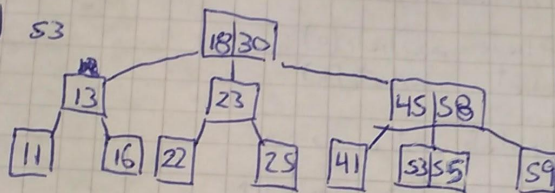
Inserto el 55



Reacomodo
y Divide



Inserto el 53



1) def generar_detalle (archivo_citas):

```
infile = open (archivo_citas, "r")  
reader = csv.reader (infile, delimiter = ";")  
diccionario = {}
```

```
for fila in reader:
```

```
    if fila[2] not in diccionario:
```

```
        diccionario[fila[2]] = ("", 0)
```

```
    diccionario[fila[2]][0] += fila[0] + fila[1] + fila[3] + fila[4] + "\n"
```

```
    diccionario[fila[2]][1] += 1
```

```
infile.close()
```

```
crear_archivo(diccionario)
```




```

def crear_archivo(diccionario):
    detalle = ""
    for obra_social in diccionario:
        detalle += obra_social + "\n" + diccionario[obra_social][0] + "\n" + obra_social + "Total"
        + diccionario[obra_social][1] + "pacientes" + "\n\n"
    outfile = open("detalle.txt", "w")
    outfile.write(detalle)
    outfile.close()

```

2) a) def construir_compras(que_se_cocina):

```

    lista_de_compras = {}
    nombre = plato[0] for (plato) in que_se_cocina:
        cantidad = plato[1]
        for (ingrediente) in recetas[nombre]:
            Nombre_ingredient = ingrediente[0]
            X cantidad_ingredient = ingrediente[1]
            if nombre_ingredient not in docena:
                if nombre_ingredient not in lista_de_compras:
                    lista_de_compras[nombre_ingredient] = 0
                lista_de_compras[nombre_ingredient] += cantidad_ingredient * cantidad
            else:
                if nombre_ingredient in lista_de_compras:
                    lista_de_compras[nombre_ingredient] += cantidad_ingredient * cantidad
                else:
                    if cantidad_ingredient * cantidad > docena[nombre_ingredient]:
                        lista_de_compras[nombre_ingredient] = cantidad_ingredient * cantidad
                        - docena[nombre_ingredient]
                        docena[nombre_ingredient] = 0
                    else:
                        docena[nombre_ingredient] -= cantidad_ingredient * cantidad
    return lista_de_compras

```

esto solo lista los índices (nombres de los platos) y la dem.

Lucas Nevada
40768

Estructura de Datos

2/2
20/9/16

b) def persistir (diccionario, archivo)
 wfile = open (archivo, "wb")
 pickle.dump (diccionario, wfile)
 wfile.close()

def recuperar (archivo):

 rfile = open (archivo, "rb")
 diccionario = pickle.load (rfile)
 rfile.close()
 return diccionario