Jose Escobar

80536060

CS2302 Lab 5 Report

Instructor: Diego Aguirre

**Introduction**

Lab 5 required to complete the given implementation of a Min-Heap data structure to implement heapsort. To make the heapsort implementation we were given options for testing the Min-Heap method

**Proposed Solution**

For this lab I decided to make two files, ‘heap’ file containing the given and finished Min-Heap implementation with its node and the other ‘test’ file that imports the heap and tests that the method worked correctly.

**Experimental Results**

The results for my lab were as expected with a few runs and debugging process to get the final results. Zybooks helped as well for the Min-Heap implementation.



**Conclusion**

I find heap data structure a fun method to implement and a structure that could heap a lot for a reasonable size array with good running times. This method helped to better familiarize with Min-Heap implementation and the way it works.

**Appendix**

'''

Jose Escobar

UTEP ID 80536060

CS2302

Lab 5A: Min heap implementation

'''

class Heap:

def \_\_init\_\_(self):

self.heap\_array = []

def insert(self, k):

self.heap\_array.append(k)

self.min\_heap()

def extract\_min(self):

if self.is\_empty():

return None

min\_elem = self.heap\_array[0]

i = 0

left = 2\*i+1

right = 2\*i+2

# Method avoids invalid/empty positions

while(right < len(self.heap\_array)):

if(self.heap\_array[left] > self.heap\_array[right]):

new\_min = self.heap\_array[right]

index = right

else:

new\_min = left

index = left

self.heap\_arrayarray[i] = new\_min

i = index

right = 2\*i+1

left = 2\*i+2

return min\_elem

def is\_empty(self):

return len(self.heap\_array) == 0

#Method sorts the array using min heap

def min\_heap(self):

if(self.is\_empty == 0):

return

for i in range(len(self.heap\_array)): #Loop traverses the heap\_array

k = len(self.heap\_array) -1

while(k >= i): #Loop swaps smaller elements than its parents

if(self.heap\_array[k-1//2] < self.heap\_array[i]):

temp = self.heap\_array[k-1//2]

self.heap\_array[k-1//2] = self.heap\_array[i]

self.heap\_array[i] = temp

k=-1

return

**Test file**

from heap import Heap

heap = Heap()

heap.insert(45)

heap.insert(8)

heap.insert(132)

heap.insert(0)

heap.insert(7)

heap.insert(98)

heap.insert(41)

heap.insert(99)

heap.insert(3)

heap.insert(4)

print(heap.heap\_array)

**Academic Honest Certification**

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.

Jose Escobar