Faculty of Engineering
Alexandria University
CSED 2024
Data Structure-2

Lab3
Implementing Binary Heap
&
SortingTechniques

Toka Ashraf Abo elwafa 19015539
Bassant Yasser Salah 19017262

1-Implementing BinaryHeap & SortingTechniques:

Problem Statement (first Requirement):

In this assignment, you're required to implement some basic procedures and show how they could be used in a sorting algorithm: The MAX-HEAPIFY procedure, which runsinO(lg n) time, is the key to maintaining the max-heap property. The BUILD-MAX-HEAP procedure, which runs in linear time, produces a max-heap from an unordered input array.

•The HEAPSORT procedure, which run sin O(nlgn) time,sorts an array inplace. The MAX-HEAP-INSERT, and HEAP-EXTRACT-MAX procedures, which run in O(lg n)time,allow the heap data structure to implement a priority queue.

Explanation (first requirement):

- **Heapify:** it just takes heap elements and compares the child with its parent in case that child is greater than parent itself then a swapping between parent and child will happen to make sure that parent is bigger than the child. Dividing array into two parts left and right.
- **Build_Max_Heap:** enter an array and its length as a parameter to build a max heap from given input. It executes the **heapify method**.
- **MaxHeapInsert:** it takes an element and compares it with the parent if it is bigger than it then we will swap them up using the **SwapUp** method.
- **Heapsort():** take an array and its size as a parameter. Build heap from a given array using the array using **Build_Max_Heap** method. Take the first element in the heap and swap twitch all other elements from the heap until it reaches the leaf. At the end the heapify method will be applied on the heap to make sure that the elements are arranged.

Test cases:

```
Enter your choice:(build- insert - max - compare):
insert
Enter the heap array (separate elements with space between them):
10 1 7 8 -5 -100 a


After insertion:
 10 8 7 1 -5 -100
Enter your choice:(build- insert - max - compare):

Enter your choice:(build- insert - max - compare):
max


The Maximum Of The Heap: 10
```

```
Enter your choice:(build- insert - max - compare):
build
Enter the heap array (separate elements with space between them):
20 18 -5 7 0 17 9 a
 20 18 17 7 0 -5 9
```

```
build
Enter the heap array (separate elements with space between them):
12 40 44 8 -450 78 a
 78 40 44 8 -450 12
Enter your choice:(build- insert - max - compare- Heapsort):

Enter your choice:(build- insert - max - compare- Heapsort):
build
Enter the heap array (separate elements with space between them):
-100 -500 a
 -100 -500
```

```
Enter your choice:(build- insert - max - compare- Heapsort):
Heapsort
Enter the heap array (separate elements with space between them):
10 -100 0 70 40 54 100 7 8 a
The Heap Array After Sorting:
 -100 0 7 8 10 40 54 70 100
```

Problem Statement (Sorting Technique):

You are required to implement the “heapsort” algorithm as an application for binary heaps. You’re required to compare the running time performance of your algorithms against:

– An O(n2) sorting algorithm such as Selection Sort, Bubble Sort, or Insertion sort.

– An O(n lg n) sorting algorithm such as Merge Sort or Quick sort algorithm in the average case.1.

In addition to heapsort, select one of the sorting algorithms from each class mentioned above.

To test your implementation and analyze the running time performance, generate a dataset of random numbers and plot the relationship between the execution time of the sorting algorithm versus the input size.

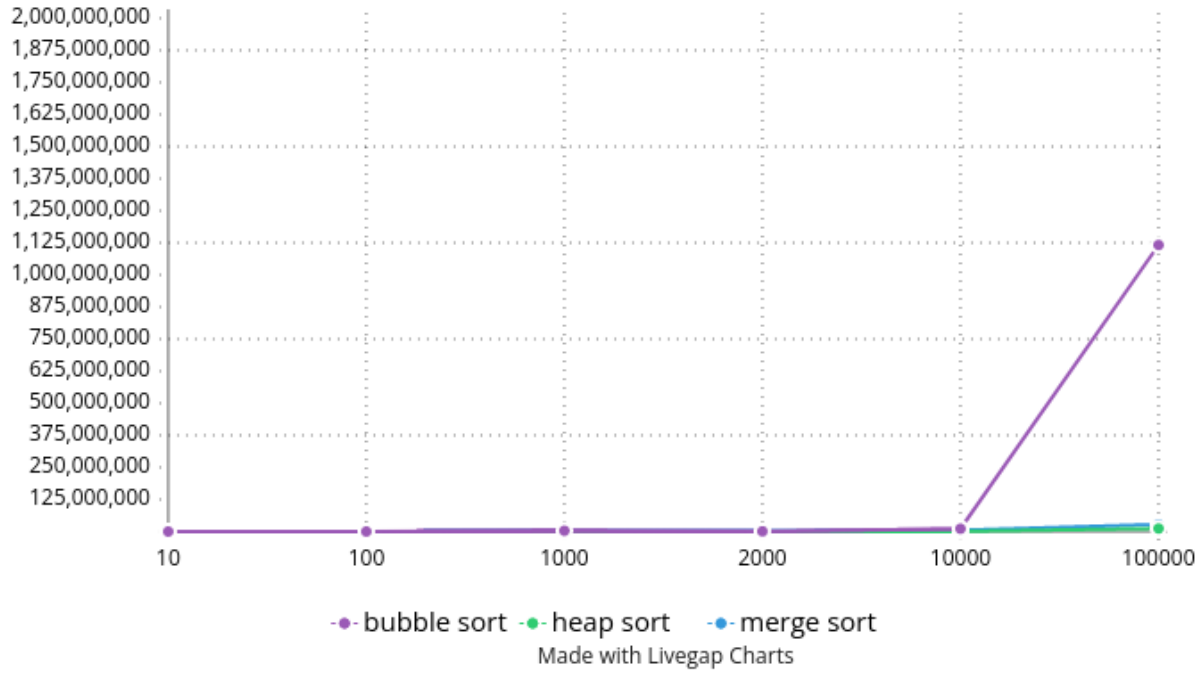
Explanation (Sorting techniques):

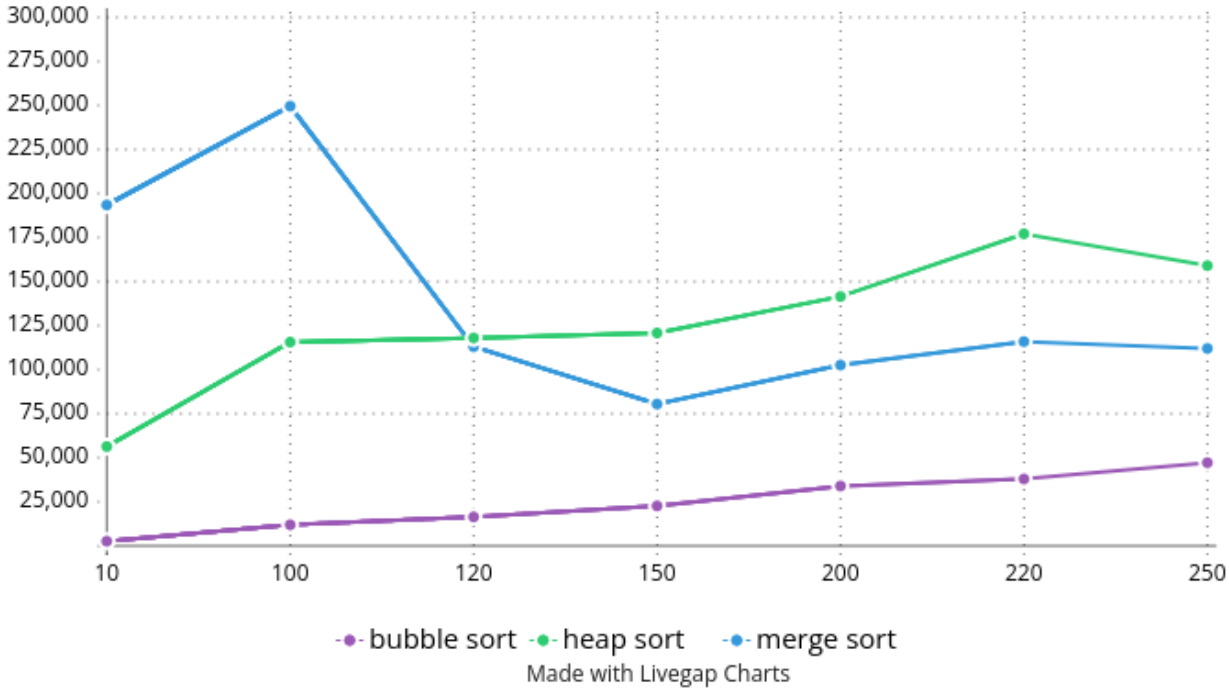
Merge sort: splitting the given array into two arrays and recursively continue to split the splitted parts until it reaches into one element only then it will start to compare elements with each other to arrange them and combine them again after each combination the elements will be compared to be arranged again.

Bubble sort: It is the simplest way to arrange elements,just compare elements rapidly and swap them until we reach an arranged order.

Graph:

input size	heapsort	mergesort	bubblesort
10	296694	20294	4818
100	100565	231330	115140
1000	192944	2414874	3099973
10000	1060959	5104262	11087526
2000	329875	1719312	832303
100000	11857669	27936265	1116550109





Test cases:

```
compare
Enter size of the array:
10
merge time=37831
MergeSort:
-----------------------------------------------------------
 -1849575203 -1237835478 -1074941288 -898827331 -658274922 626906575 724676413 876822147 1732240923 1839581161
bubble time=4855
Bubblesort:
-----------------------------------------------------------
 -1849575203 -1237835478 -1074941288 -898827331 -658274922 626906575 724676413 876822147 1732240923 1839581161

HEAP_SORT time=221142
HEAP_SORT:
-----------------------------------------------------------
 -1849575203 -1237835478 -1074941288 -898827331 -658274922 626906575 724676413 876822147 1732240923 1839581161
```

```
Enter your choice:(build- insert - max - compare):
compare
Enter size of the array:
10
merge time=46048
bubble time=9511


HEAP_SORT time=446138
```

```
Enter your choice:(build- insert - max - compare):
compare
Enter size of the array:
100
merge time=248929
bubble time=379856


HEAP_SORT time=348961
```

```
Enter your choice:(build- insert - max - compare):
compare
Enter size of the array:
1000
merge time=2148985
bubble time=10694026


HEAP_SORT time=339337
```

```
Enter your choice:(build- insert - max - compare):
compare
Enter size of the array:
10000
merge time=4858119
bubble time=13039157


HEAP_SORT time=1209652
```

How to use:

1- The user has 4 choices to choose between (insert- build-compare-max -HeapSort):

1) Compare: if it is used then it means that comparing in the runtime between heap sort- merge sort -Bubble sort and this happens randomly as the input array is randomized.
2) Build: build a heap max from input array. Users enter only an array to build it in a max heap.
3) Insert: take an array from the user and perform an operation like insert but the only difference is insert build heap array while build not.
4) Max: get max element from the heap.

2- According to the user's choice, the method will be performed.

Assumptions:

1- while entering an array in any method from the following(insert-build), you must enter the elements in one line and the spaces between them, at the end of the line insert a character to end the insertion process.

Ex:

15 5 10 -100 80 40 a.