

DESIGN.pdf ASGN3

Nathan Ong

October 14, 2021

1 Description

This collection of files contains four sorting methods that can be implemented to sort through an array of size n . The testing rig used to test these sorting methods is also usable with the following command line options:

- -a: Runs all sorting algorithms
- -e: Runs the heap sort method
- -i: Runs the insertion sort method
- -s: Runs the shell sort method
- -q: Runs the quick sort method
- -r *seed*: Sets the RNG seed to the argument *seed*. The default seed is 13371453.
- -n *size*: Sets the array size to the argument *size*. The default size is 100 elements.
- -p *elements*: Prints out *elements* number of elements from the array. The default number is 100. If the array is smaller than the argument, then the entire array is printed.
- -h: Displays a help message detailing the usage of the program

2 Files Included in the Directory

1. insert.c

- (a) This file contains the source code for the implementation of the insertion sorting algorithm.

2. insert.h

- (a) This file contains the specification of the interface to insert.c.

3. heap.c

- (a) This file contains the source code for the implementation of the heap sorting algorithm.

4. heap.h

- (a) This file contains the specification of the interface to heap.c.

5. quick.c

- (a) This file contains the source code for the implementation of the quick sorting algorithm.

6. quick.h

- (a) This file contains the specification of the interface to quick.c.

7. set.h

- (a) This file contains the source code for the implementation and the specification of the interface to the set ADT (abstract data type).

8. stats.c

- (a) This file contains the implementation of the statistics module listed in stats.h.

9. stats.h

- (a) This file contains the specification of the interface to the statistics module in stats.c.

10. shell.c

- (a) This file contains the source code of the implementation of the shell sorting algorithm.

11. shell.h

- (a) This file contains the specification of the interface to shell.c.

12. sorting.c

- (a) This file contains the main function of the interface in addition to any additional functions that are necessary for the implementation of the assignment.

3 Pseudocode and Structure

3.1 insert.c

```
for the length of the array
  store array element temporarily
  while there is still elements to be sorted
    swap elements of position k and position k-1
  store element at k-1 temporarily
return sorted array
```

3.2 heap.c

```
for the length of the array
  build a heap from the array
  fix the heap (make it a max heap)
  lock position of the largest element in the heap
return sorted array
```

3.3 quick.c

```
for the length of the array
  split array in half with pivot element
  move anything greater than pivot to the right of it
  repeat until sorted
return sorted array
```

3.4 shell.c

```
for the length of the array
  get gap between elements
  compare elements, swap if needed
  proceed and keep decreasing gap until gap is 1, then use insertion sort
return sorted array
```

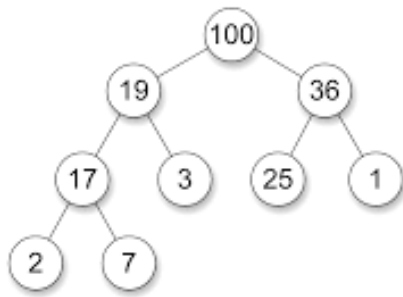
3.5 sorting.c

while opt isnt -1
 indice through arguments
 check for required arguments if necessary
 use set to enable functions to be performed
 generate array to be sorted and perform required functions

4 Additional Diagrams

4.1 Heap Sort

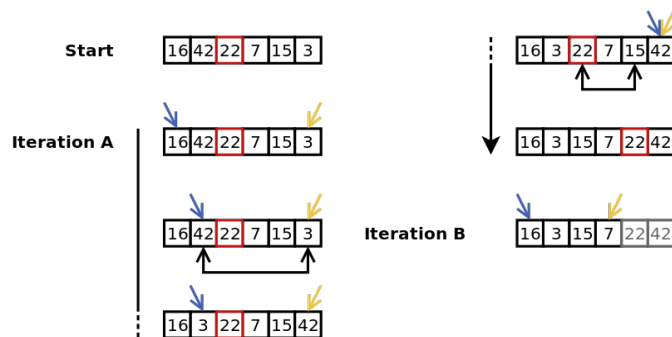
Tree representation



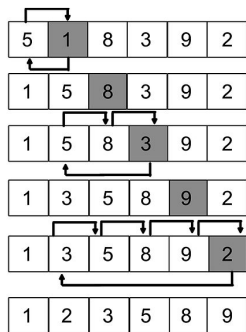
Array representation



4.2 Quick Sort



4.3 Insertion Sort



5 Additional Credits

- Sloan's section on October 5th provided the Makefile formatting.
- Images use Creative Commons License CC BY-SA 3.0.