Christina Choi

Milla Penelope Markovic

Data Structures: Homework 5

1. See <https://github.com/cchoi17/LMU-CMSI2120/tree/main/Homework5>

2. a(4,2)

= a(3, a(4,1))

= a(3, 65533)

= 2^{65536}- 3

= 2 ↑↑ 5 - 3

3. Worst case

a). Recursive: 2^n

b). Non-recursive: 2n but in **Θ** notation: **Θ**(n)

4.) [81 9 17 21 20 8 2 5 1 83 23]

a.) Selection Sort

81 9 17 21 20 8 2 5 1 83 23

1 81 9 17 21 20 8 2 5 83 23

1 2 81 9 17 21 20 8 5 83 23

1 2 5 81 9 17 21 20 8 83 23

1 2 5 8 81 9 17 21 20 83 23

1 2 5 8 9 81 17 21 20 83 23

1 2 5 8 9 81 17 21 20 83 23

1 2 5 8 9 17 81 21 20 83 23

1 2 5 8 9 17 20 81 21 83 23

1 2 5 8 9 17 20 21 81 83 23

1 2 5 8 9 17 20 21 23 81 83

1 2 5 8 9 17 20 21 23 81 83

1 2 5 8 9 17 20 21 23 81 83

b.) Gnome Sort

81 9 17 21 20 8 2 5 1 83 23

9 81 17 21 20 8 2 5 1 83 23

9 81 17 21 20 8 2 5 1 83 23

9 17 81 21 20 8 2 5 1 83 23

9 17 81 21 20 8 2 5 1 83 23

9 17 81 21 20 8 2 5 1 83 23

9 17 81 21 20 8 2 5 1 83 23

9 17 21 81 20 8 2 5 1 83 23

9 17 21 81 20 8 2 5 1 83 23

9 17 21 81 20 8 2 5 1 83 23

9 17 21 81 20 8 2 5 1 83 23

9 17 21 20 81 8 2 5 1 83 23

9 17 21 20 81 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

9 17 20 21 8 81 2 5 1 83 23

9 17 20 21 8 81 2 5 1 83 23

9 17 20 8 21 81 2 5 1 83 23

9 17 20 8 21 81 2 5 1 83 23

9 17 8 20 21 81 2 5 1 83 23

9 17 8 20 21 81 2 5 1 83 23

9 8 17 20 21 81 2 5 1 83 23

9 8 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

8 9 17 20 21 2 81 5 1 83 23

8 9 17 20 21 2 81 5 1 83 23

8 9 17 20 2 21 81 5 1 83 23

8 9 17 20 2 21 81 5 1 83 23

8 9 17 2 20 21 81 5 1 83 23

8 9 17 2 20 21 81 5 1 83 23

8 9 2 17 20 21 81 5 1 83 23

8 9 2 17 20 21 81 5 1 83 23

8 2 9 17 20 21 81 5 1 83 23

8 2 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 8 9 17 20 21 5 81 1 83 23

2 8 9 17 20 21 5 81 1 83 23

2 8 9 17 20 5 21 81 1 83 23

2 8 9 17 20 5 21 81 1 83 23

2 8 9 17 5 20 21 81 1 83 23

2 8 9 17 5 20 21 81 1 83 23

2 8 9 5 17 20 21 81 1 83 23

2 8 9 5 17 20 21 81 1 83 23

And continue…

c.) Insertion Sort

81 9 17 21 20 8 2 5 1 83 23

9 81 17 21 20 8 2 5 1 83 23

9 17 81 21 20 8 2 5 1 83 23

9 17 21 81 20 8 2 5 1 83 23

9 17 20 21 81 8 2 5 1 83 23

8 9 17 20 21 81 2 5 1 83 23

2 8 9 17 20 21 81 5 1 83 23

2 5 8 9 17 20 21 81 1 83 23

1 2 5 8 9 17 20 21 81 83 23

1 2 5 8 9 17 20 21 81 83 23

1 2 5 8 9 17 20 21 23 81 83

d.) Quick Sort

81 9 17 21 20 8 2 5 1 83 23

2 5 1 9 8 17 81 21 20 83 23

2 5 1 9 8 17 81 21 20 83 23

1 2 5 9 8 17 21 20 23 81 83

1 2 5 8 9 17 20 21 23 81 83

…

81 9 17 21 20 8 2 5 1 83 23

81 9 17 21 20 8 2 5 1 23 83

23 9 17 21 20 8 2 5 1 81 83

23 9 17 21 20 8 2 5 1 | 81 83

1 9 17 21 20 8 2 5 23 | 81 83

1 9 17 21 20 8 2 5 23 | 81 83

1 9 5 21 20 8 2 17 23 | 81 83

1 8 5 2 9 20 21 17 23 | 81 83

1 2 5 8 9

e.) Merge Sort

81 9 17 21 20 8 2 5 1 83 23

81 9 17 21 20 8 2 5 1 83 23

[81 9] [17 21] [20 8] [2 5] [1 83] 23

[9 81] [17 21] [8 20] [2 5] [1 83] 23

[9 17 21 81] [8 20 2 5] [1 83 23]

[9 17 21 81] [2 5 8 20] [1 83 23]

[9 17 21 81 2 5 8 20] [1 23 83]

[2 5 8 9 17 20 21 81] [1 23 83]

1 2 5 8 9 17 20 21 23 81 83

f.) Heap Sort (Heapify, Swap, Remove)

81 9 17 21 20 8 2 5 1 83 23

83 9 17 21 20 8 2 5 1 81 23 (Heapify)

23 9 17 21 20 8 2 5 1 81 83 (Swap)

23 9 17 21 20 8 2 5 1 81 83 (Remove)

81 9 17 21 20 8 2 5 1 23 83 (Heapify)

23 9 17 21 20 8 2 5 1 23 83 (Swap)

23 9 17 21 20 8 2 5 1 81 83 (Remove)

23 9 17 21 20 8 2 5 1 81 83 (Heapify)

1 9 17 21 20 8 2 5 23 81 83 (Swap)

1 9 17 21 20 8 2 5 23 81 83 (Remove)

21 9 17 1 20 8 2 5 23 81 83 (Heapify)

5 9 17 1 20 8 2 21 23 81 83 (Swap)

5 9 17 1 20 8 2 21 23 81 83 (Swap)

20 9 17 1 5 8 2 21 23 81 83 (Heapify)

2 9 17 1 5 8 20 21 23 81 83 (Swap)

2 9 17 1 5 8 20 21 23 81 83 (Remove)

And continue…

g.) Radix Sort (Sort by least to greatest digit)

81 9 17 21 20 8 2 5 1 83 23

20 81 21 1 2 83 23 5 17 8 9 (Sorted into buckets based on ones place)

1 2 5 8 9 17 20 21 23 81 83 (Iterating through buckets, sorting into buckets based on tens place.)

5.) Sentences describing each.
Exchange: Compares the first element in the array with the following elements, swapping ones out of order.
Insertion: Beginning at the first element, iterating through each following element, sliding elements back to its proper relative position.
Selection: Iterating through the array, comparing the first element to each following element of the array, finding the smallest element and swapping them.
Merge: Break down the array in half until you have singleton lists, then merge the elements into sorted sublists, until finally you have one sorted list.
Distribution: spreads things out, and then sweep through and gather everything in order
Hybrid: A mix of different sorting algorithms, a bit of merge sorting, quick sort, etc.
Concurrent: Sorting by accessing the left and right array simultaneously.
Impractical: Sorting algorithms where you try things and see if it works.

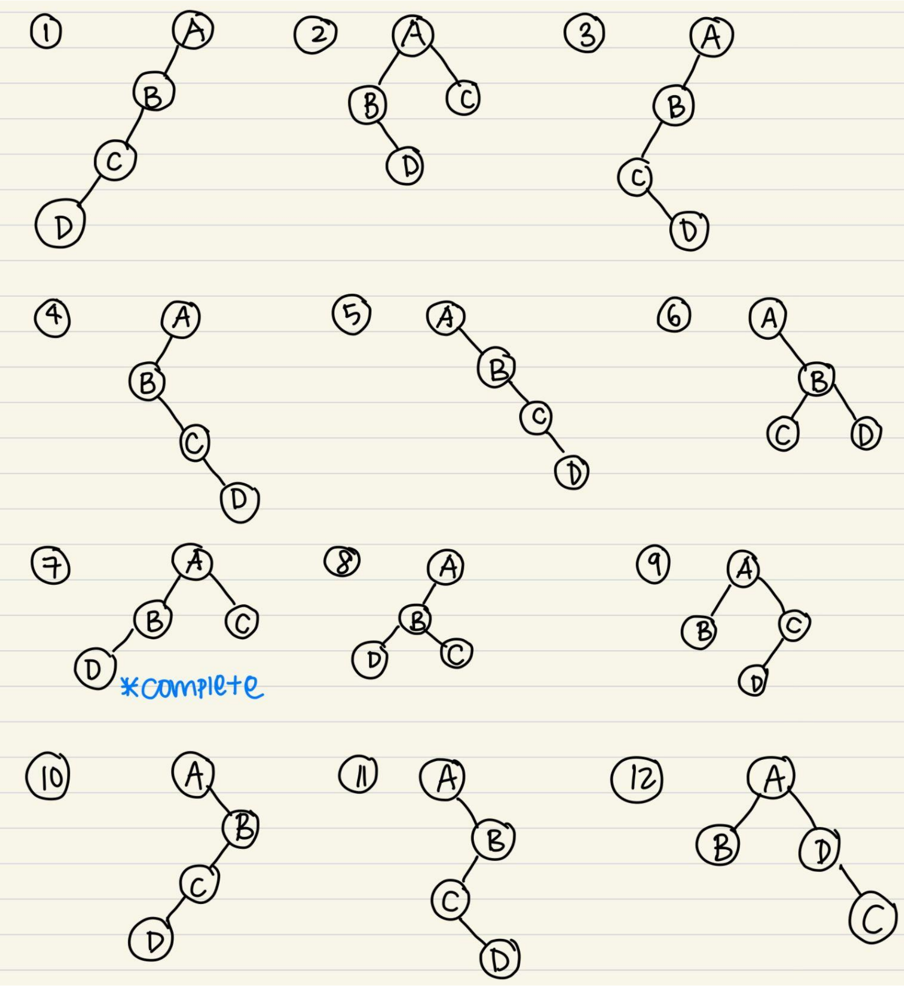
6.) See<https://github.com/cchoi17/LMU-CMSI2120/tree/main/Homework5>

7.) For the following tree:

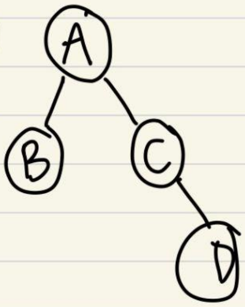
A. What is the size of this tree? 16
B. How many edges does this tree have? 15
C. What are the leaves? C, F, H, J, L, M, N, O, P
D. What are the children of D? G
E. What is the depth of G? 2
F. What is the degree of G? 4

G. What are the ancestors of G? D, A
H. What are the descendants of G? J, K, L, M, O, P
I. What are the nodes on level 3? H, I, J, K, L, M
J. What is the height of the tree? 4
K. What is the width of the tree? 6
L. What is the height of the node D? 3
M. What is the simplest path from P to E? P, K, G, D, A, B, E
N. What is the degree of the tree? 15
O. Enumerate the nodes in breadth-first order. A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P
P. Enumerate the nodes in depth-first order. A, B, E, H, I, N, F, C, D, G, J, K, O, P, L, M

8.) Draw all 14 binary trees of size 4. Circle the ones that are complete trees.



13



14