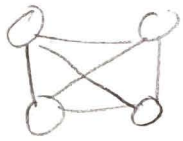
EECS 233 SI Session
Leader: Bertram Su
December 5, 2019

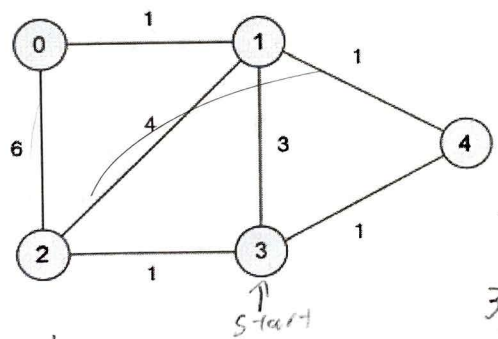


Objectives:
Upon completion of this SI session, participants will be able to:
1. Find the minimum spanning tree from Prim's algorithm
2. Understand the uses for MST

Foundations:
1. What is a minimum spanning tree (MST)?
   A subset of edges that connects all the edges w/ the least total cost
2. Comment the steps for Prim's algorithm.
   a) A[] stores visited, C[v] stores distance from parent, p[v]stores parent
   b) add starting vertex u to A * heap
   c) for all nodes v
      if v adjacent to u
         C[v] = c(u,v)
         p[v] = u
      else C[v] = infinity
   d) Loop until all nodes are in A
      a. find w not in A with min cost
      b. add w to A
      c. update C[v] for all v adjacent to w and not in A
         C[v] = min(C[v], c(w, v))
         update P[v] if C[v] changed

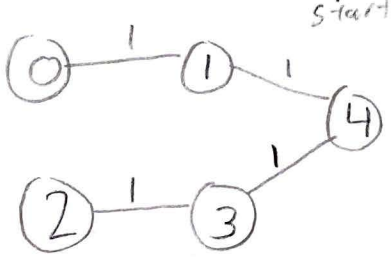
// set up is same as Dijkstras
// same dijstra
// different

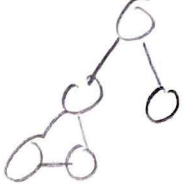
3. Determine a MST that starts from vertex 3. If costs are the same go 1 → 2 → 3 →…



A	0	1 ✓	2 ✓	4 ✓
3	∞, null	3, 3	1, 3	1, 3
3, 2	6, 2	3, 3		1, 3
3, 2, 4	6, 2	1, 4		
3, 2, 4, 1	1, 1			

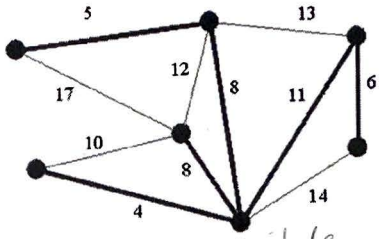
3, 2, 4, 1, 0



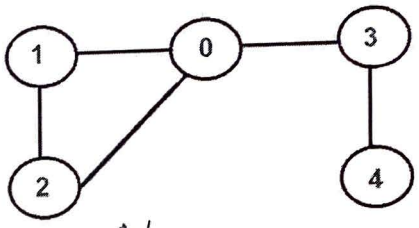


Exercises:

4. Which of the following is a possible MST?

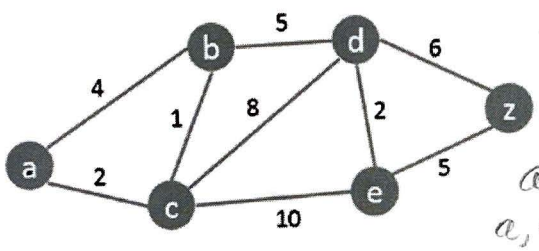


possible



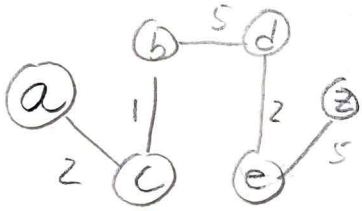
No, cycle

5. Using Prim's algorithm, find a minimum spanning tree. Start from a



A	b ✓	c ✓	d ✓	e ✓	z
a	4, a	2, a	∞	∞	∞
a, c	1, c		8, c	10, c	∞
			5, b	10, c	∞
				2, d	6, d
					5, e

a, c, b
a, c, b, d
a, c, b, d, e
a, c, b, d, e, z



6. Rudolph wants to visit z as fast as possible because the kids wake up early. What algorithm would you use to get the result?

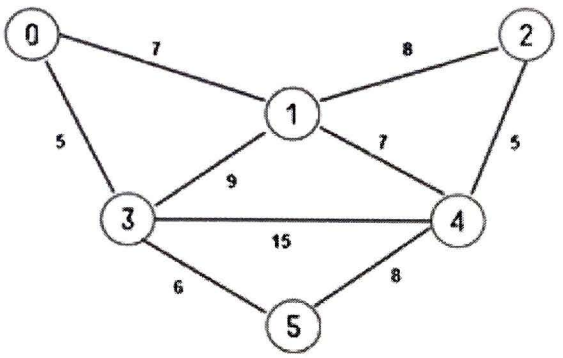
Dijkstra

7. What's the runtime of Prim's algorithm?

E log V

Summary

8. Create a MST For the follow tree start from 3



9. You've decided to pack up your bags and work for Santa to give hot chocolate to the elves. You're given a map, a hot chocolate source, and distances to houses and are told to find the most efficient way to build pipes to each house. What algorithm would you recommend?

prim

10. Dasher decided to eyeball the map and claims he came up with an MST. Prancer decided to use your recommended algorithm and claims he came up with a <u>different</u> MST. Is one of them guaranteed to be wrong? No, you can have different

11. Which student(s) is/are correct?

Dasher: Dijkstra's algorithm gives you the most efficient path to visit all nodes in a single trip ✗
Comet: Dijkstra's algorithm gives you the most efficient path from an origin vertex to another ✓
Vixen: Dijkstra's algorithm gives you headaches. ~~
Blitzen: Prim's algorithm gives you the lightest edge cost to connect all nodes ✓

Thanks for a great semester!

Upcoming Events and Suggestions for Further Study:

Events:

- Next SI session is Sunday from 1:00 to 2:30 at Nord 204 [FINAL REVIEW]

Further Study:

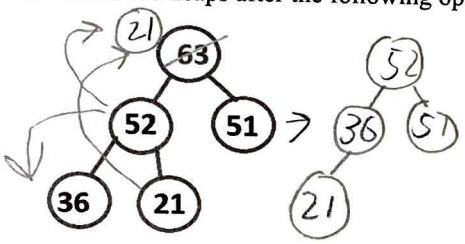
- <u>https://cwru.az1.qualtrics.com/jfe/form/SV_1Th0sizrbealYXz</u>
  - Survey Link
- <u>https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/</u>
  - Some code for prim's algorithm

EECS 233 SI Session
Leader: Bertram Su
December 8, 2019

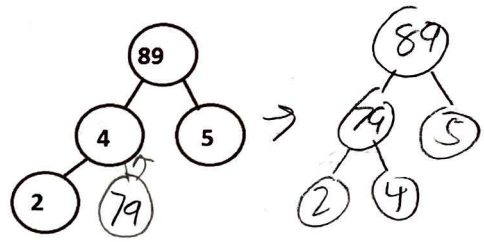
Objectives:
Upon completion of this SI session, participants will have reviewed the major concepts from the second half of Introduction to Data Structures.

Heaps

1. Draw the heaps after the following operations

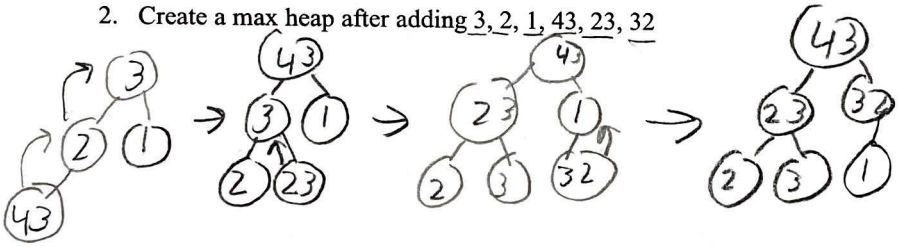


Remove



Add 79

2. Create a max heap after adding 3, 2, 1, 43, 23, 32



3. What are the left child, right child, and parent indexes of a node at index i?

Parent = $\frac{i-1}{2}$

Left = $i \times 2 + 1$

Right = $i \times 2 + 2$

4. Draw the first original tree from 1 in its array form

0	1	2	3	4
63	52	51	36	21

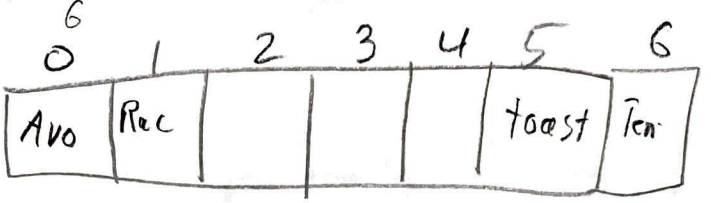
5. What is the big O runtime of add, remove, and creating a heap?

Add : logn
remove : logn

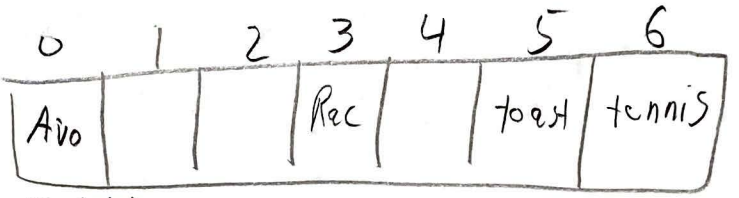
create : n

Hashing

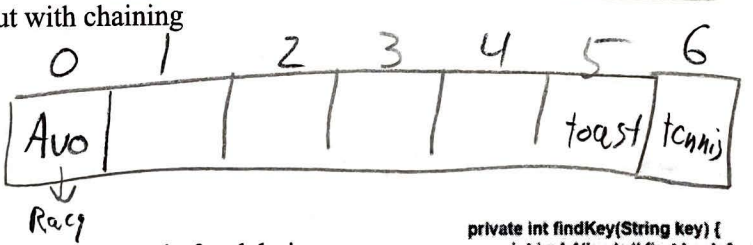
6. Using a hash function of string length % 7, draw the array after inserting "Avocado", "Toast", "Racquet", "Tennis". If a collision happens, use linear probing



7. Draw the array after inserting the same input as above but use double hashing. h1(key) = length % 7 and h2(key) = key % 4



8. Solved but with chaining



9. Finish the following code for deletion

```
public void delete(int key)
        int i = findKey(key)
        if(i == -1)
                return;
        else table[i].removed = true;

}
```

```
private int findKey(String key) {
    int i = h1(key); // first hash function
    int j = h2(key); // second hash function
    int iterations = 0;

    // keep probing while the entry is not empty

    while ( table[i] != null                    ){
        // return if key is found, otherwise continue
        if (table[i].removed==false && table[i].key.equals(key))
            return i;

        i = (i + j) % tableSize;
        iterations++;
        if (iterations >= tableSize) return -1;
    }

    return -1;
}
```

10. What is the big O of add, remove, and search for hash tables?

Add : n
remove : n
search : n

11. Why do we have a removed Boolean?

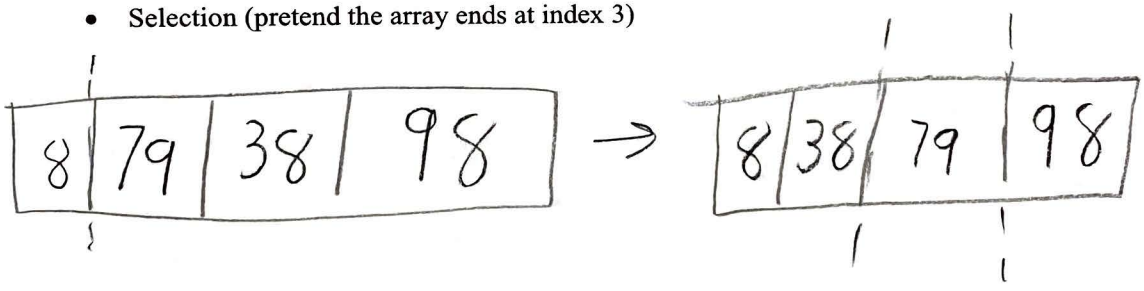
continue searching
inserting is faster

Sorting

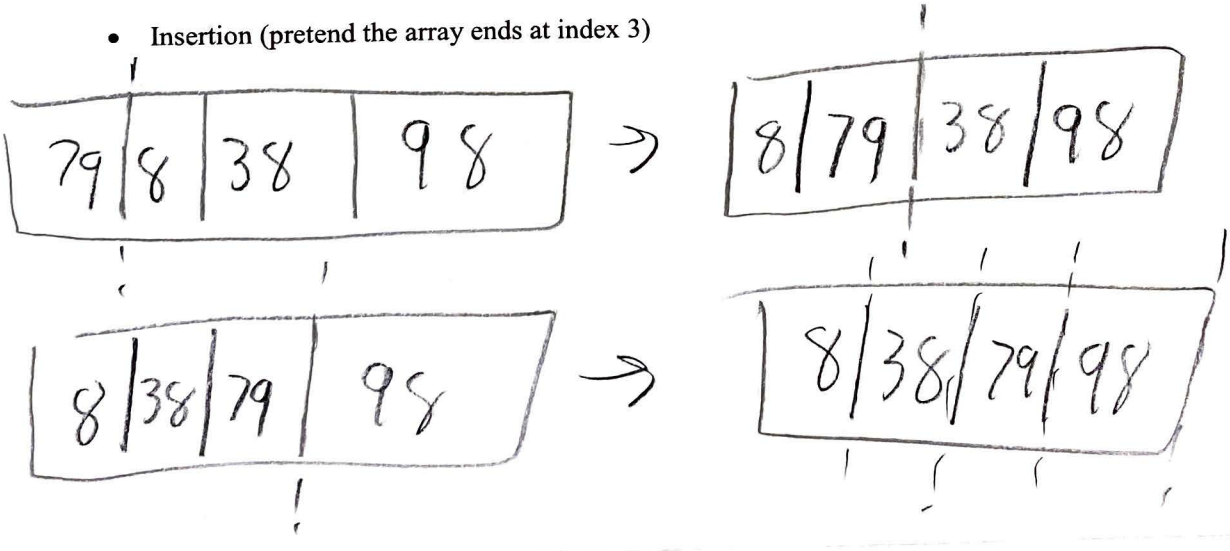
12. Draw the array phases for each sorting method with the following array

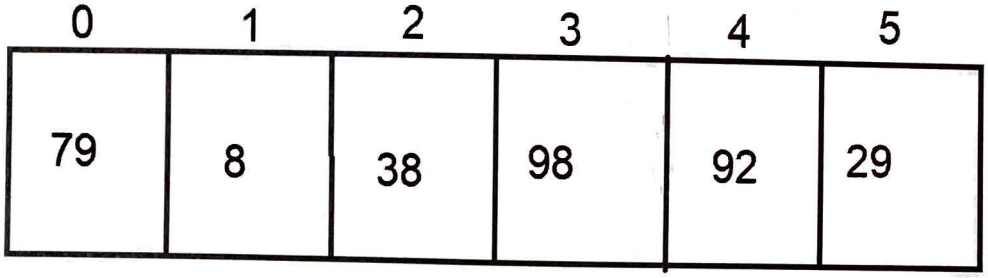
0	1	2	3	4	5
79	8	38	98	92	29

- Selection (pretend the array ends at index 3)

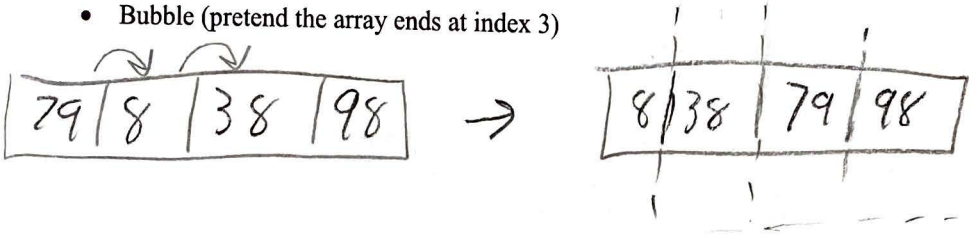


- Insertion (pretend the array ends at index 3)





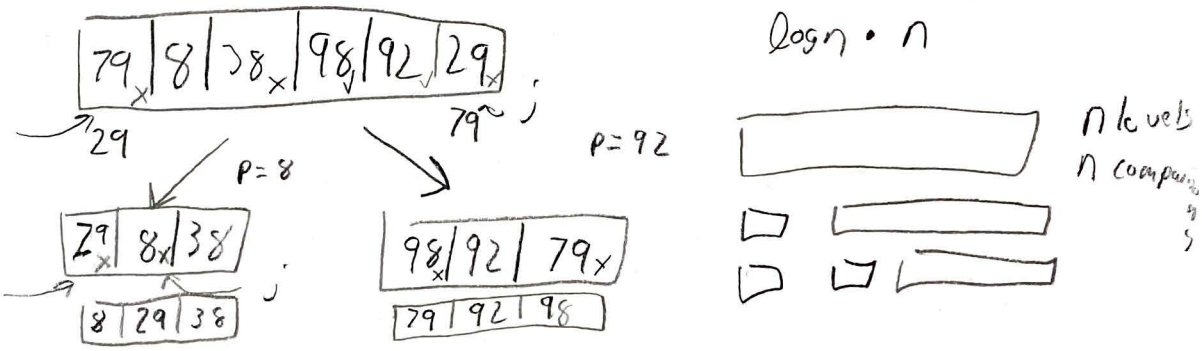
- Bubble (pretend the array ends at index 3)



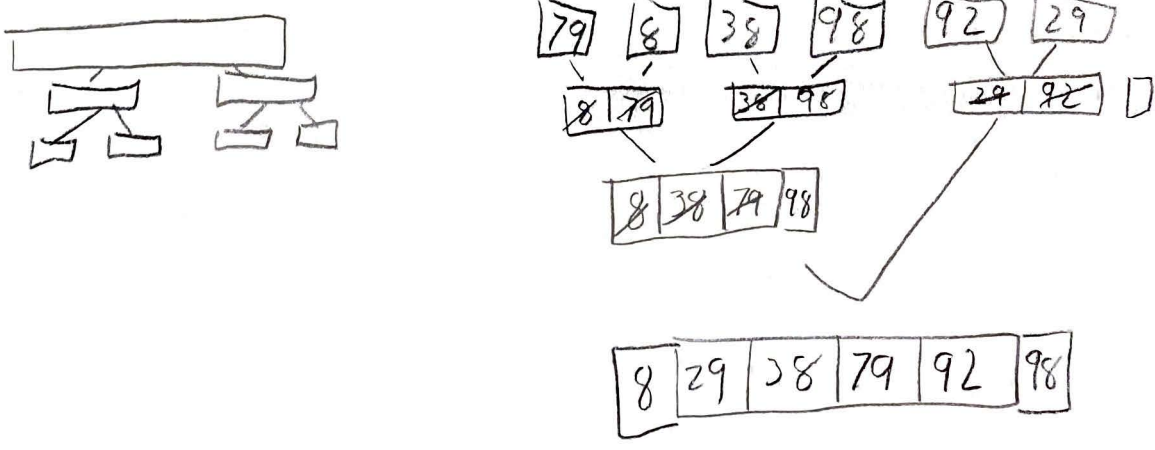
- Quick (middle pivot) p = 38

levels comparisons

log n • n



- Merge

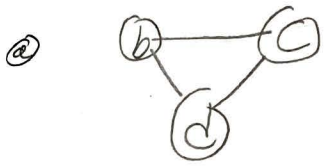


13. Fill in the following chart

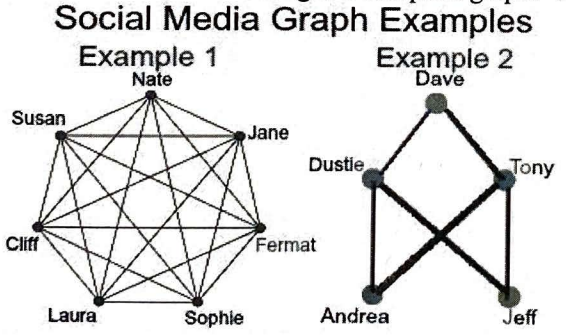
Sort	Best-Case	Worst-Case
Selection	n^2	n^2
Bubble	opt n n^2	n^2
Insertion	n	n^2
Quick	$n \log n$	n^2
Merge	$n \log n$	$n \log n$
Heap	$n \log n$	$n \log n$

Graphs

14. Draw a graph that is not connected, contains a cycle, and is undirected



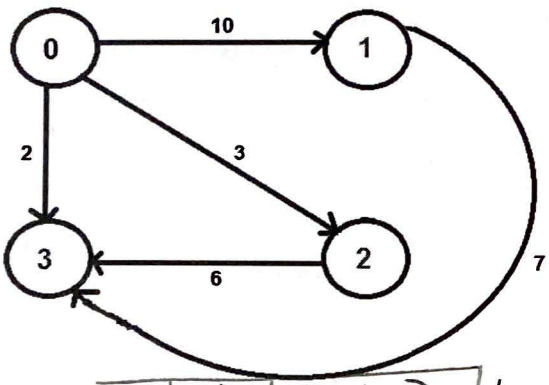
15. Which of the following is a complete graph? Which is only a connected graph?

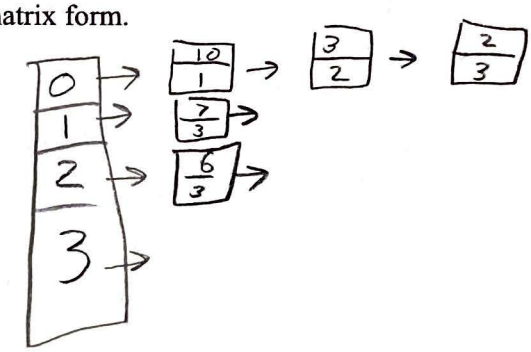


Complete

Connected

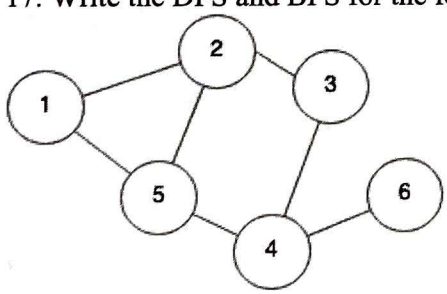
16. Draw the following graph in its adjacency list and matrix form.





from \ to	0	1	2	3
0	0	10	3	2
1		0		7
2			0	6
3				0

17. Write the DFS and BFS for the following graph

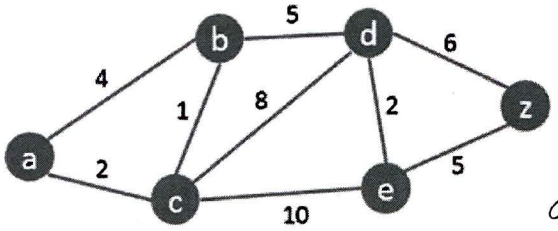


DFS Stack: ~~1, 2, 3, 4, 5, 6~~
visited: 1, 2, 3, 4, 5, 6
print: 1, 2, 3, 4, 5, 6

BFS Q: ~~1, 2, 5, 3, 4, 6~~
visited: 1, 2, 5, 3, 4, 6
print: 1, 2, 5, 3, 4, 6

Dijkstra's and Prim's Algorithm

18. Finish Dijkstra's and Prim's for the following graph

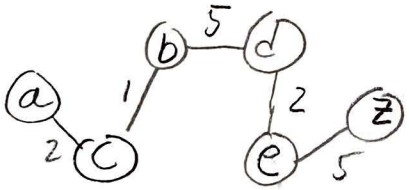


Visited	b ✓	c ✓	d ✓	e ✓	z
					∞
a	4, a	2, a	∞	∞	∞
a, c	3, c		10, c	12, c	∞
a, c, b			8, b	12, c	∞
a, c, b, d				10, d	14, d
a, c, b, d, e					14, d
a, c, b, d, e, z					

A → d

d, b, c, a

Visited	b ✓	c ✓	d ✓	e ✓	z
					∞
a	4, a	2, a	∞	∞	∞
a, c	1, c		8, c	10, c	∞
a, c, b			5, b	10, c	∞
a, c, b, d				2, d	6, d
a, c, b, d, e, z					5, e



19. What is the runtime for Dijkstra's and Prims for the array implementation?

$E \log V$

Upcoming Events and Suggestions for Further Study:

Events:

- I heard there's a final

Further Study:

- REVIEW THE QUIZZES
- bigocheatsheet.com
  - A great graph that visualizes the big o complexity chart. It also has the big O time of data structures and algorithms that we will cover in the future.