CSCE 221 Cover Page

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more Aggie Honor System Office https://aggiehonor.tamu.edu/

Name	Mualla Argin
UIN	728003004
Email address	margin25@tamu.edu

Cite your sources using the table below. Interactions with TAs and resources presented in lecture do not have to be cited. Please remove any sources you did not use.

People	1. Not Applicable
Webpages	1. https://www.youtube.com/watch?v=g9YK6sftDi0
Printed Materials	1. None
Other Sources	1. None

Homework 3

Due April 27 at 11:59 PM

Typeset your solutions to the homework problems preferably in LaTeXor LyX. See the class webpage for information about their installation and tutorials.

1. (10 points) An airport is developing a computer simulation of air-traffic control that handles events such as landings and takeoffs. Each event has a *time-stamp* that denotes the time when the event occurs. The simulation program needs to efficiently perform the following two fundamental operations:

1. Insert an event with a given time-stamp (that is, add a future event)
2. Extract the event with a smallest time-stamp (that is, determine the next event to process)

(a) What data structure should be used to implement the above operations efficiently? Explain your reasoning.

The Priority Queue data structure should be used because it stores elements that are sorted based on a factor (i.e. timestamp is a type of factor). Priority Queues keep a tree-like structure and elements are put in it according to their priority.

(b) Provide the big-O asymptotic complexity of inserting a time-stamp into the data structure identified in part (a)

If we use a minimum heap the component is first put toward the end. From that point forward, it is contrasted and its parent, in the event that it is smaller than its parent, its moved one level up in the tree. In this case the big O asymptotic complexity for inserting is $\log(n)$.

(c) Provide the big-O asymptotic complexity of extracting a time-stamp from the data structure identified in part (a) The elements are removed in O(1) but It takes O(log n) to adjust the heap. If we took the element with the minimum timestamp, we need to look for the minimum timestamp element in the remaining heap to make the tree min-heap again. In this case the big O asymptotic complexity for extracting is $\log(n)$.

2. (10 points) Draw a 17-entry hash table that results from the using the hash function: $h(k) = ((3k + 5) \mod 11)$ to hash the keys: 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5. Assume collisions are handled using the double hashing method. The secondary hash function is given by $h_s(k) = (7 - (k \mod 7))$.

Provide the final state of the table below:

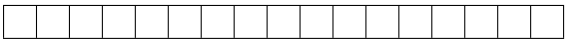


Table 1: Resulting Hash Table

Provide a record of all indexes which are probed (locations where algorithm inserts or attempts to insert a key). Note when collisions occur:

`insert(12)`:

`insert(44)`:

`insert(13)`:

`insert(88)`:

`insert(23)`:

`insert(94)`:

`insert(11)`:

`insert(39)`:

`insert(20)`:

`insert(16)`:

`insert(5)`:

Inserting 12
main hash value is 8
12 is inserted at position 8
-, -, -, -, -, -, -, -, 12, -, -, -, -, -, -, -, -

Inserting 44
main hash value is 5
44 is inserted at position 5
-, -, -, -, -, 44, -, -, 12, -, -, -, -, -, -, -, -

Inserting 13
main hash value is 0
13 is inserted at position 0
13, -, -, -, -, 44, -, -, 12, -, -, -, -, -, -, -, -

Inserting 88
main hash value is 5
There is already an item in 5
double hash value is 3
5 + 1*3 = 8
So, checking at index 8 There is already an item in 8
double hash value is 3
5 + 2*3 = 11
So, checking at index 11 88 is inserted at position 11
13, -, -, -, -, 44, -, -, 12, -, -, 88, -, -, -, -, -

Inserting 23
main hash value is 8
There is already an item in 8
double hash value is 5
8 + 1*5 = 13
So, checking at index 13 23 is inserted at position 13
13, -, -, -, -, 44, -, -, 12, -, -, 88, -, 23, -, -, -

Inserting 94
main hash value is 1
94 is inserted at position 1
13, 94, -, -, -, 44, -, -, 12, -, -, 88, -, 23, -, -, -

Inserting 11
main hash value is 5
There is already an item in 5
double hash value is 3
5 + 1*3 = 8
So, checking at index 8 There is already an item in 8
double hash value is 3
5 + 2*3 = 11
So, checking at index 11 There is already an item in 11
double hash value is 3
5 + 3*3 = 14
So, checking at index 14 11 is inserted at position 14
13, 94, -, -, -, 44, -, -, 12, -, -, 88, -, 23, 11, -, -

Inserting 39
main hash value is 1
There is already an item in 1
double hash value is 3
1 + 1*3 = 4
So, checking at index 4 39 is inserted at position 4
13, 94, -, -, 39, 44, -, -, 12, -, -, 88, -, 23, 11, -, -

Inserting 20
main hash value is 10
20 is inserted at position 10
13, 94, -, -, 39, 44, -, -, 12, -, 20, 88, -, 23, 11, -, -

Inserting 16
main hash value is 9
16 is inserted at position 9
13, 94, -, -, 39, 44, -, -, 12, 16, 20, 88, -, 23, 11, -, -

Inserting 5
main hash value is 9
There is already an item in 9
double hash value is 2
9 + 1*2 = 11
So, checking at index 11 There is already an item in 11
double hash value is 2
9 + 2*2 = 13
So, checking at index 13 There is already an item in 13
double hash value is 2
9 + 3*2 = 15
So, checking at index 15 5 is inserted at position 15
13, 94, -, -, 39, 44, -, -, 12, 16, 20, 88, -, 23, 11, 5, -

HashTable

---

0 - 13
1 - 94
2 -
3 -
4 - 39
5 - 44
6 -
7 -
8 - 12
9 - 16
10 - 20
11 - 88
12 -
13 - 23
14 - 11
15 - 5
16 -

3. (15 points) A *complete* graph is an undirected graph in which every pair of vertices are connected with an edge. Consider the following complete graph with $n = 6$ vertices.

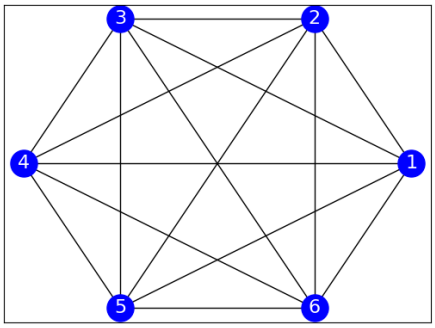


Figure 1: A complete graph with $n = 6$ vertices

(a) In what order does `DFS` explore vertices in the above graph? Assume `DFS` starts at vertex `4`. The adjacency lists are in ascending order by the numeric label. According to the question , the traversal will start at vetex 4. Where we start the traversal with vertex 4 then we have (n-1) option to select the next vertex in our case. There is 5 options . Let the first vector and the vector 1 ,Now after this we will have 4 options to selct from vertex. We will select vertex 2 then we will have 3 options and so on

(b) What is the running time of `DFS` on a complete graph with n vertices? Provide an asymptotic big-oh bound in terms of the number of vertices. Explain your reasoning. The time complexity of a DFS traversal is O(V + E). If your graph is implementing using adjacency list where in each node maintains a list of all its adjacent then, for each node we could discover all of its neighbors by traversing its adjacency list just once in linear time for a directed graph. the sum of the sizes of the adjancency list of all nodes iE. So the complexity is O(V) + O(E) = O(V + E).

(c) How many back edges, forward edges, cross edges, and exploratory edges are generated by running `DFS` on a complete graph with n vertices? DFS has tree with degree 1 or 2 only.So, it has just back edges and exploratory edges.undirected graphs don't have forward edges.

4. (10 points) Answer each of the following questions with a tight big-oh asymptotic bound. Justify with algorithmic reasoning.

(a) A priority queue, `UnsortedMPQ`, is implemented based on an unsorted array. What is the running time of the operation which retrieves the minimum value?

Priority Queues are similar to queues.We add an element from the back and eliminate an element from the front.Since the Priority Queue is unsorted we would have to go through the whole array to get the time complexity. Hence the time complexity is O(n).

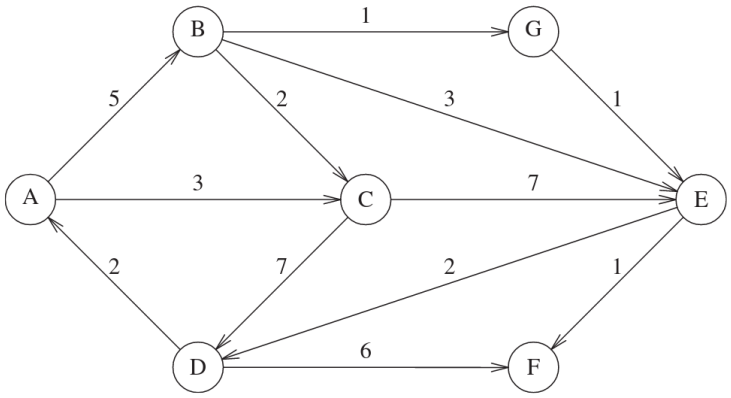
$$\texttt{unsorted\_min(n)} \in O(\cdots) \tag{1}$$

(b) Dijkstra's algorithm is implemented based on this unsorted minimum priority queue. What is the running time of this implementation of Dijkstra's algorithm?

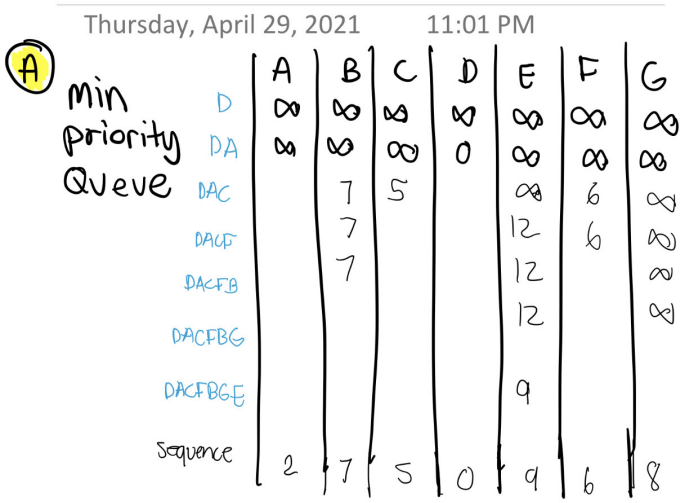
Every time the main loop executes, one vertex from the queue is removed. Assuming the graph has n vertices, the queue could have O(n) number of vertices. each pop takes O(log n) time. As a result, the total time taken to perform the main loop is O (n log n).the time spent in the function extend is considered. Since expand is called only once per vertex, handle edge is called only once per edge.When push() is called, there can only be n calls during the execution, so the overall expense is (n log n). In conclusion, the total run time is O(n log n + n log n).

$$\texttt{dijkstra(V, E)} \in O(\cdots) \tag{2}$$

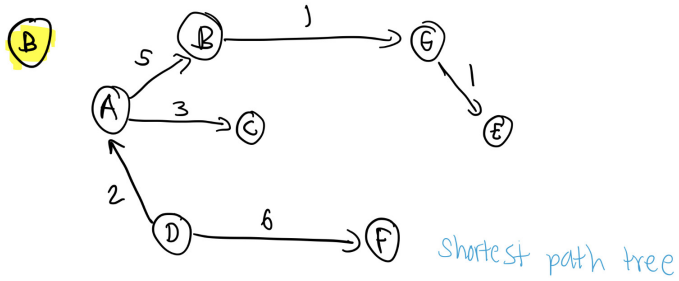
5. (15 points) Find the shortest path from D to all other vertices for the graph below.



(a) Illustrate the minimum priority queue at each iteration Dijkstra's algorithm.



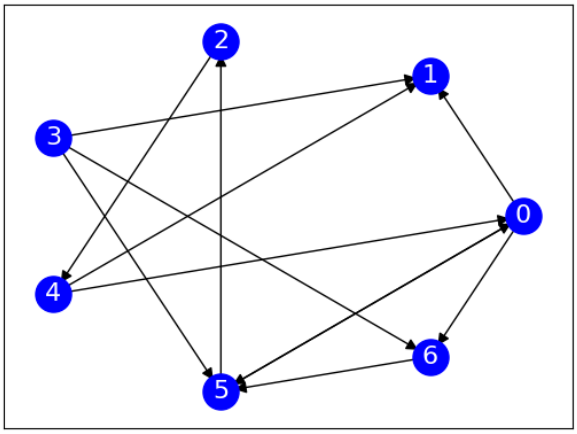
(b) Draw the Shortest Path Tree.



(c) What is the running time of the Dijkstra's algorithm under the assumption that the graph is implemented based on an adjacency list and the minimum priority queue is implemented based on a binary heap?

The running time of the Dijkstra's algorithm is O ((V+E)log(V))

6. (15 points) Find the shortest path from vertex 3 to all other vertices for the graph below.



(a) Which graph algorithm can solve the problem most *efficiently*? Dijsktras shortest path finding algorithm can solve the problem efficiently

(b) Draw the Shortest Path Tree.

Thursday, April 29, 2021 11:16 PM

Shortest path length is 2: 3→5→0

Shortest path length is 1 : 3→1

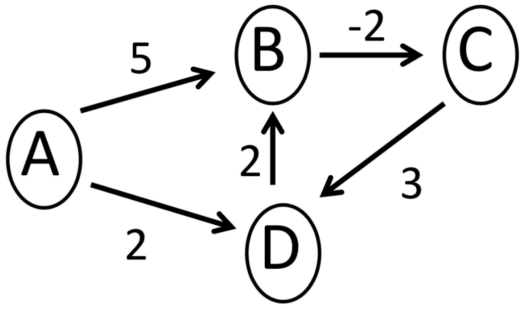
Shortest path length is 2 : 3→5→2

Shortest path length is 3 : 3→5→2→4

Shortest path length is 1 : 3→5

Shortest path length is 1 : 3→6

7. (10 points) Apply the Dijkstra's algorithm to find the shortest path from the vertex A to all the vertices in the graph below.



(a) Illustrate the minimum priority queue at each iteration Dijkstra's algorithm. 1 - ABCD
2 - DBC
3 - BC
4 - C
5 - gamma

(b) Does the algorithm return the correct output? Yes, the algorithm returns the correct output through the method of Induction

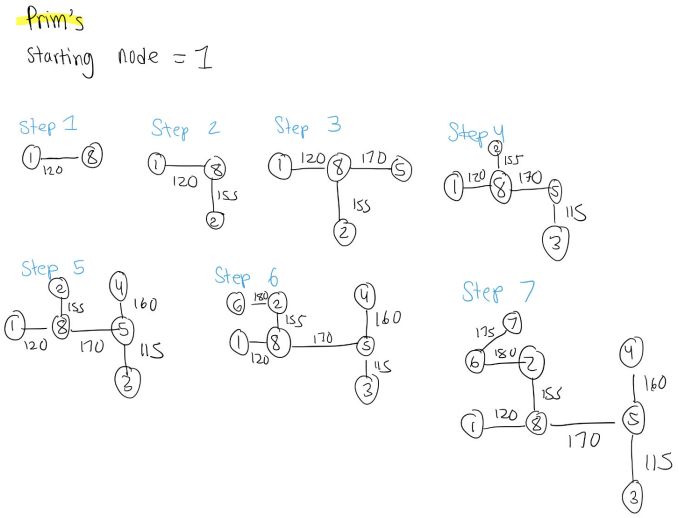
(c) Does Dijkstra's Theorem guarantee correctness for this graph? Explain. Yes, Dijkstra's Algorithm is correct for this graph because the graph doesn't contain any negative cycle. The negative cycle doesn't fit for Dijsktra's theorem because if it encounters a negative cycle it will into a infinite loop as each iteration will keep reducing the path cost.

8. (15 points) There are eight small island in a lake, and the state wants to build seven bridges to connect them so that each island can be reached from any other one via one or more bridges. The cost of bridge construction is proportional to its length. The distance between pairs of islands are given in the following table.

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	120
2	-	-	265	175	215	180	185	155
3	-	-	-	260	115	350	435	195
4	-	-	-	-	160	330	295	230
5	-	-	-	-	-	360	400	170
6	-	-	-	-	-	-	175	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

Table 2: The distance between any two islands

1. Illustrate the Prim's algorithm using the graph below. Draw the Minimum Spanning Tree. What is the length of the bridges?

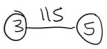


2. Illustrate the Kruskal's algorithm using the graph below. Draw the Minimum Spanning Tree. What is the length of the bridges?

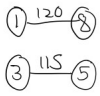
Thursday, April 29, 2021 11:16 PM

Kruskal's Algorithm

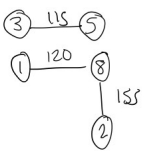
Step 1



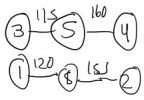
Step 2



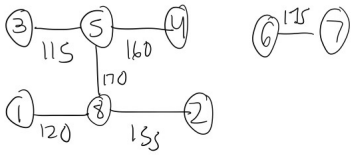
Step 3



Step 4



Step 5



Step 6