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	Chase Albright
UIN	529008060
Email address	chasealbright@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. Grigori Rasputin
Webpages	1. https://doi.org/10.1901/jaba.1974.7-497a
Printed Materials	1. None
Other Sources	1. None

Homework 3

Due April 28th 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. (15 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.

A min heap / priority queue, this is the best data structure because we want to queue up the upcoming flight times this makes the most sense for this situation

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

Big o = O(log(n))

(c) Provide the big-O asymptotic complexity of extracting a time-stamp from the data structure identified in part (a)

Big o = O(log(n))

2. (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.

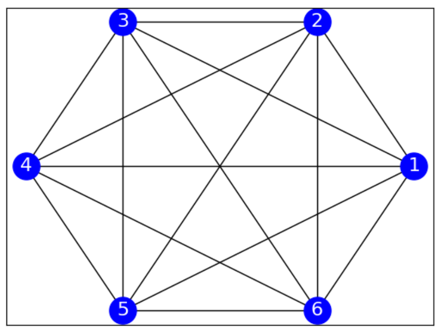


Figure 1: A complete graph with $n = 6$ vertices

(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.

$4 -> 1 -> 2 -> 3 -> 5 -> 6$

(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.

In a complete graph edges(e) $= n(n-1)/2$ and the big O is O(n+e) simplified,
this would be $O(n + (n^2 - n)/2 = O(n + n^2)$ which reduced leads to a Big-O $= O(n^2)$ this is because of the interconnectedness of all the vertices

(c) How many back edges, forward edges, cross edges, and exploratory edges are generated by running `DFS` on a complete graph with n vertices?

There are no cross edges in a complete graph
n-1 exploratory edges
forward and back = n(n-1)/2

3. (15 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?

This algorithm only requires a single loop bases on n size to find the min value (n)

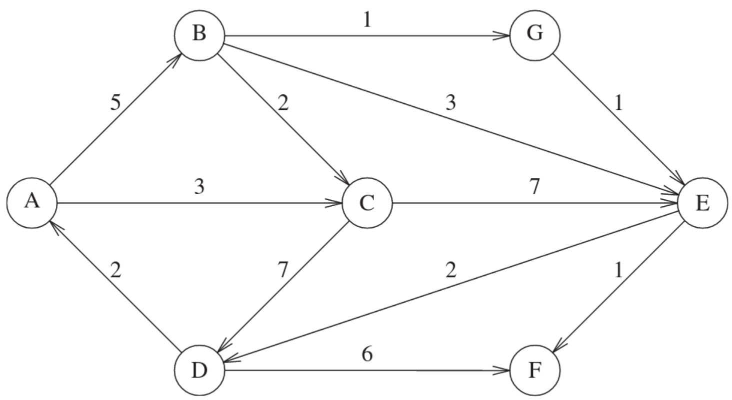
$$\texttt{unsorted\_min(n)} \in O(n) \tag{1}$$

(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?

This is because you while loop through each vertex and at each vertex you find the min vertex while looping through again (v * v), then loop through the edges (+ E)

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

4. (20 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.

4A

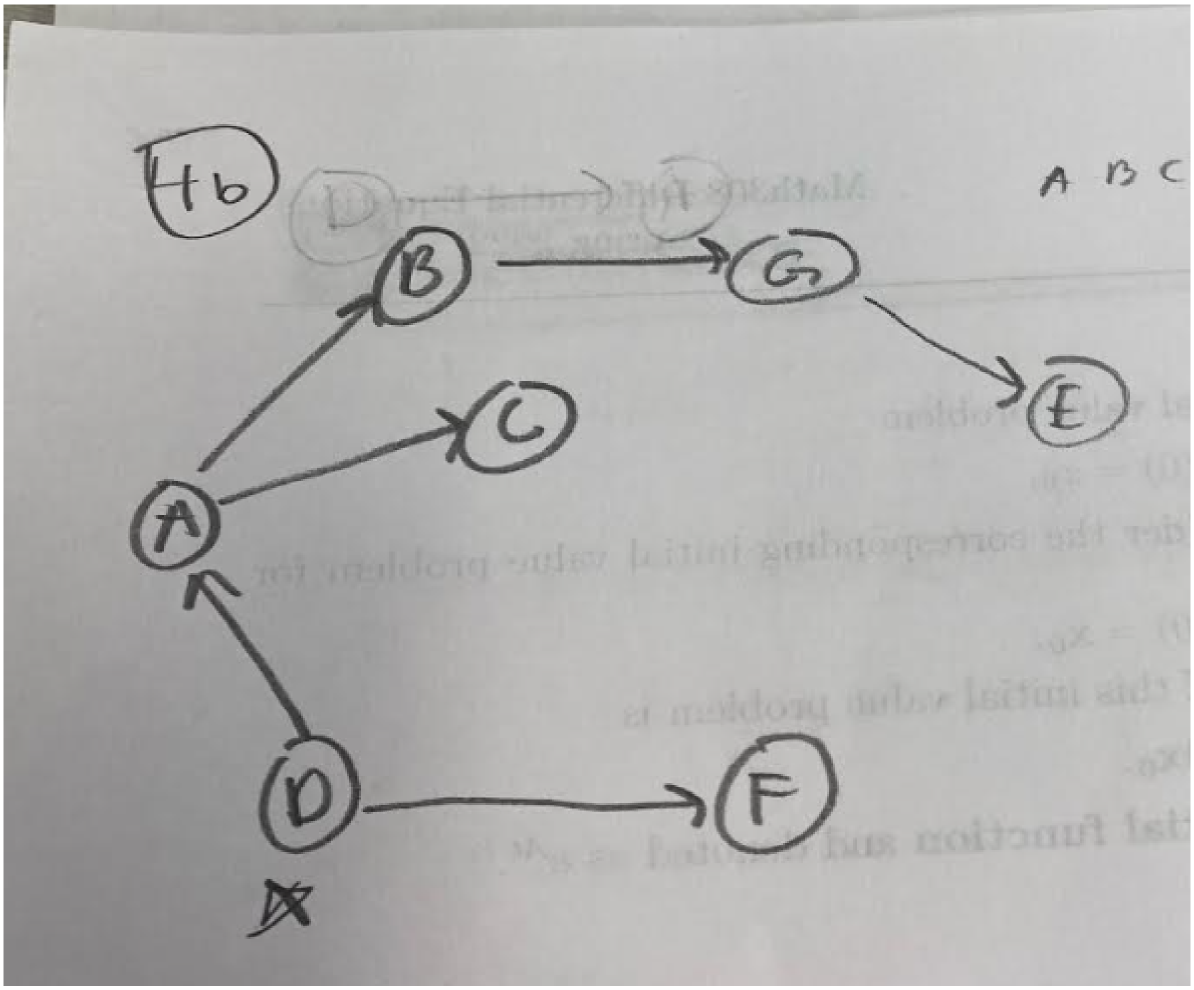
minque

D
DA
DAC
DACF
DACFB
DACFBG
DACFBGE

A	B	C	D	E	F	G
∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	$\boxed{0}$	∞	∞	∞
$\boxed{2}$	∞	∞		∞	6	∞
	7	$\boxed{5}$		∞	6	∞
	7			12	$\boxed{6}$	∞
	$\boxed{7}$			12		$\boxed{8}$
				12		
				$\boxed{9}$		

2 7 5 0 9 6 8

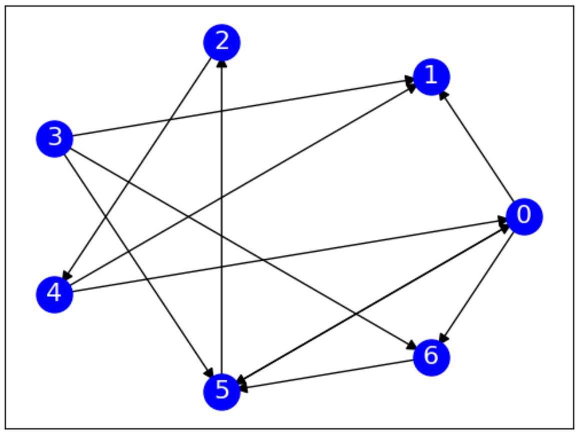
(b) Draw the Shortest Path Tree.



(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?

Running time = O((V+E)log(V))

5. (15 points) Find the shortest path from vertex 3 to all other vertices for the graph below.

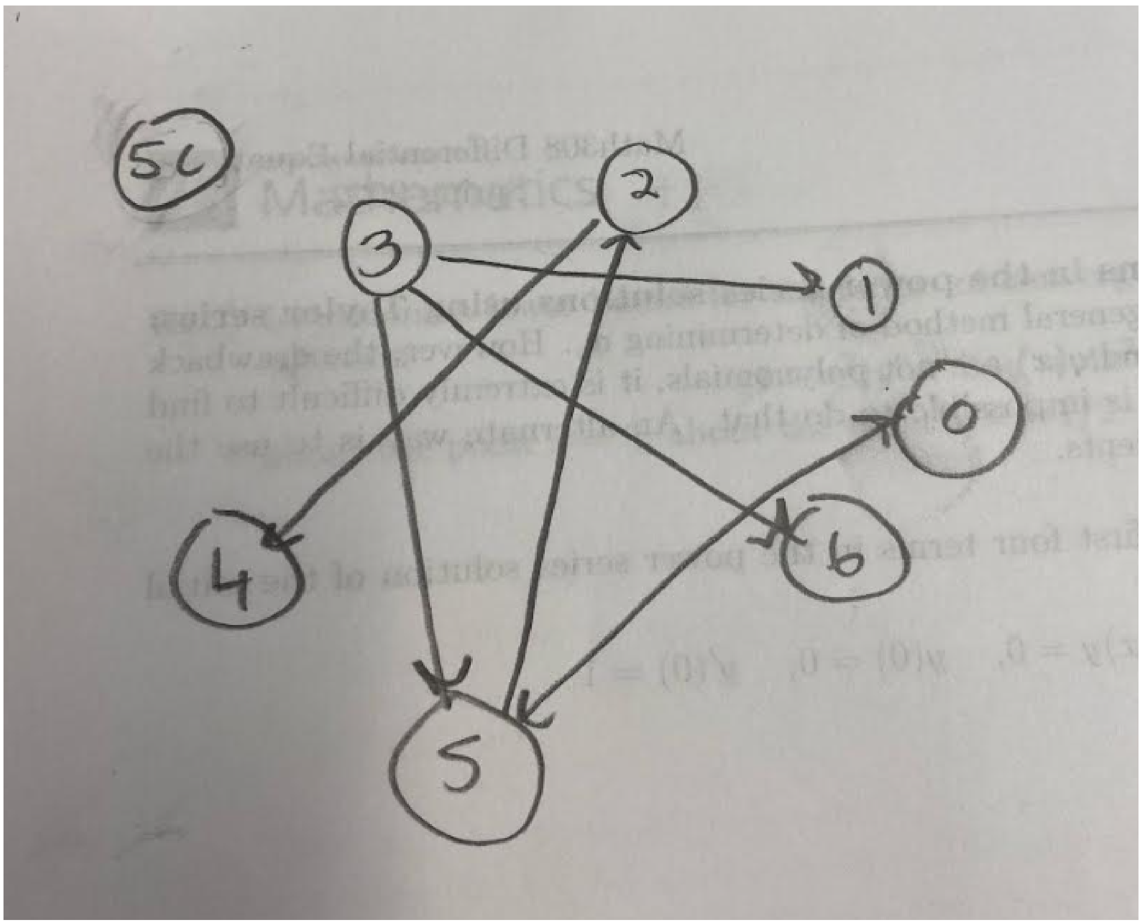


(a) Which graph algorithm can solve the problem most *efficiently*?
BFS

(b) How could you use the same algorithm if the graph had edge weights? (*Hint*: You may want to create intermediate nodes.)

After using an algorithm to create intermediate nodes (this will make the graph unweighted) then use BFS

(c) Draw the Shortest Path Tree.



6. (20 points) There are five small islands 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
1	-	10	15	10	20
2	-	-	15	20	20
3	-	-	-	15	30
4	-	-	-	-	10
5	-	-	-	-	-

Table 1: The distance between any two islands

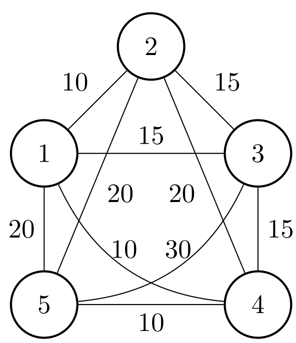
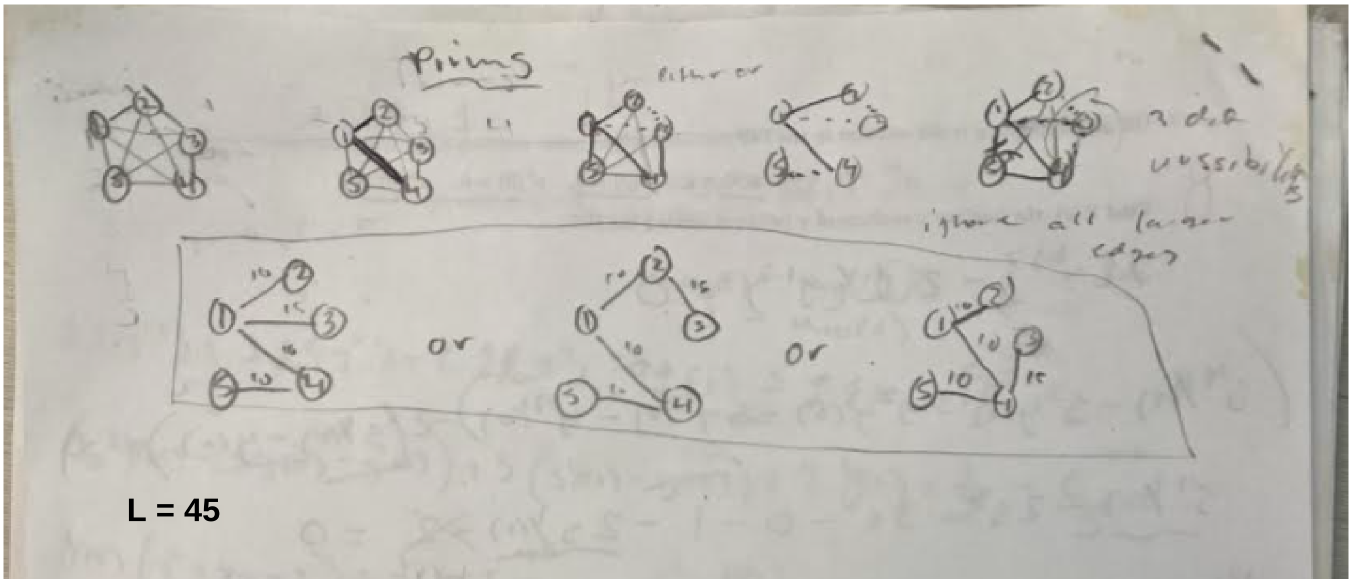


Figure 2: The distance between any two islands

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



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