CSCE 221 Cover Page
Homework Assignment #3
Due April 26 to CSNet

First Name: Josh **Last Name:** Zschiesche **UIN:** 523000614

User Name: Jzschiesche1 **E-mail address:** jzschiesche1@tamu.edu

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 http://aggiehonor.tamu.edu/

Type of sources	Peer TA Helpdesk		
People	Erik Kiggergard Todd Christian	Mason Becker	Ryan Yantz
Web pages (provide URL)	http://goo.gl/qytZqL http://goo.gl/9wmiem	http://goo.gl/7sMHCg Stack overflow	
Printed material	Slides Book		
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Josh Zschiesche Date 4/26/2016

Homework 3

due April 26

1. (10 points) R-9.10 p. 417

   What is the result of Exercise R-9.7, when collisions are handled by double hashing using the secondary hash function $h_s(k) = 7 - (k \bmod 7)$?

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

2. (10 points) R-8.2 p. 361

   How long would it take to remove $\lceil log\, n \rceil$ smallest elements from a heap that contains n entries using the `removeMin()` operation?

   $\lceil (log_2(n))^2 \rceil$

3. (10 points) R-8.7 p. 361

   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:

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

   Which data structure should be used for the above operations? Why?

   ANS:: The answer is priority que because it always pulls out the next flight in the que, relative to the minimum, meaning it stays sorted relative to the minimum. Also, if a flight needs to leave half an hour early, it can do that because there is a time key that will easily allow the programmer to modify the que and still have it sorted.

4. (10 points) R-13-3 and R-13-4, p. 654

   Look in the file AnswersHomework3Jzschiesche1.pdf.

   On 13.3 I accidently left off 1->2->3->4->5->6->7->8->1->8->7->D6->D5

   On 13.5 I accidentaly left off 1->2->3->4->5->6->7->8->9->8->7->6->5->4->3->2->1

5. (10 points) R-13.8, p. 655

   (a) 10,000 vertices, 20,000 edges, it is important to use as little space as possible ::: Adjacent list Structure: Using adjacency matrix will take up too much space. This could be fixed by making and populating a 10,000 by 10,000 matrix. The sdjsncency lidt will make 20,000 Edge nodes

   (b) 10,000 vertices and 20,000,000 edges, it is important to use as little space as possible ::: Adjacency matrix strucutre: The adjacency matrix structure stores edges in a boolean that indicates whether an edge is present between vertices that correspond to the row and the cell.

   (c) Is AdjacentTo as fast as possible, using infinite space? It only takes one operation to check if any one verticie is connect to any other verticies, meaning that this is as fast as possible.

6. (10 points) R-13.16, p. 656

   Dijkstra's algorithm is a greedy algorithm, which means that it does local optimization. With help from the priority queue, you can output a tree based on the final output of the priority queue. For example, if you have the output queue | -/0 | A/5 | A/6 | B/8 | B/9 | C/10 | C/12 | The tree can become (inset image) and you output the tree from the based on the final output, constructing the search weight from the priority queue.

7. (10 points) R-13.17, p. 656

   [1215, (240, 210, 175, 115, 180, 175, 120)]

8. (10 points) You want to help CS/CSE freshman students to prepare their course schedules for the first two years in the lower level division. By building a directed graph suggest order in which they should schedule their courses taking into account their corresponding prerequisites. A set of vertices represents courses and a set of edges represents a dependence of a given course on a course prerequisite.

   ANS:: Look in the file AnswersHomework3Jzschiesche1.pdf.

9. (10 points) R-13.31, p. 656

   ANS: Every internal node in the tree only has one parent and one child, with the exception of the root, because the root still has only one child, but has no parent(s). This is the only node with no parent, but the last node will lack children.

10. (10 points) Write what the running time, and provide its justification, of the Dijkstra's algorithm is for a sparse and dense graph and the priority queue implemented based on

    (a) a binary heap

    ANS:: Look in the file AnswersHomework3Jzschiesche1.pdf.

    (a) an unsorted list

    ANS:: Look in the file AnswersHomework3Jzschiesche1.pdf.

    (a) a sorted list

    ANS:: Look in the file AnswersHomework3Jzschiesche1.pdf.

11. (10 points) C-13.10, p. 658

    ANS:: Basically, we would have to modify depth first search so that it has a way to cheack the number of times it has visited a node. At a point where there is a distinguished vertex, which has an odd number of untraversed edges adjacent to it. If a Depth First Search is run, and not all edges have been traversed yet, we backtrack and find the last vertex we saw that has untraversed edges, making the vertex a distingushed vertex. As soon as we do that, we can attach the old circuit to the new circuit.

12. (10 points) C-13.15, p. 659

    ANS:: Look in the file AnswersHomework3Jzschiesche1.pdf.

13. (10 points) C-13.18, p. 659

    While the graph isn't empty

    1. Remove maximum element.
    2. If the vertex is at the end, stop, else continue.
    3. For every adjacent vertex, find the largest edge value, there should only be one
    4. If the edge value is larger, that one.
    5. Repeat until the edge is found.
    6. Return value.