Problem Set 6

> **Due dates:** Electronic submission of the pdf file of this homework is due on **10/22/2021 before 11:59pm** on canvas. The paper needs to be typeset in LaTeX. Handwritten solutions will receive 0 points.

Name: [redacted]

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Signature: [redacted]

Read chapter 22 in our textbook before attempting to answer these questions.

Problem 1 (20 points)**.** Solve Exercise 22.1-5 on page 593 of our textbook.

Solution. For problem 22.1-5, we must describe an efficient algorithm for computing G^2 from G for adjacency lists and matrix, and analyze run times. So considering the adjacency matrix and taking a square, would leave us with edges for each pair of vertices. These can be deciphered by a path of 2. Furthermore, in order to get a square of the graph then there must be vertices added in, which can be separated by one edge. This is when u, v are in either matrix $G^2[u, v]$ or $G[u, v]$ are 1. Matrix multiplication can be used to find the matrix and it will take approximately $O(n^2)$.

For an adjacency list, we can take the transpose of the graph, calling it T. Then G^2 will be added, so then we can parse through the list for every vertex, let's take v. We can then see if it is in u, and then add u to the appropriate v, and add it to v's list in the new graph, T. This run time can possibly add up to $O(|E||V| + |V|)$, or $O(n^3)$ where n is the length of $|V|$.

Problem 2 (20 points)**.** Solve Exercise 22.1-7 on page 593 of our textbook.

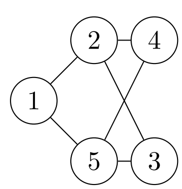
Solution. For problem 22.1-7, we must describe the entries of matirx BB^T represent, where B^T is the transpose of B given

$b_{ij} = -1$ if edge j leaves vertex i
$b_{ij} =\ \ 1$ if edge j enters vertex i
$b_{ij} =\ \ 0$ otherwise

So, we can say that $a_{jk} = a^T_{kj}$ and $b_{ij} = \sum_k a_{ik} * a_{jk}$, where $a_{ik} * a_{jk}$ is equal to 1. This is only if the vertices i and j are in incident with edge k, else it would be 0. In the case that, f i does not equal j, then it will count the number of edges from vertex i to j.
So, it will look like : $a_{ij} =$ to ...
a_{ij}, if i != j or a_{ii}, if i = j. Furthermore, we can say that in a_{ij}, the number of edges incident go to both V_i and V_j and in a_{ii}, the number of edges incident go to V_i.

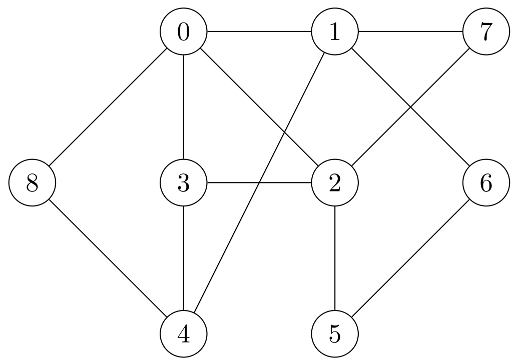
Problem 3 (20 points)**.** Solve Exercise 22.2-6 on page 602. Use tikz to draw the graph in your LaTeX document.

Solution. For problem 22.2-6, we must show an example of a directed graph.



Considering the graph above, let that graph be G, and let G' be the same graph without vertices from 3-5 and 2-4, where (V, E_π). In both graphs 1 is a source vertex. In order to show that E_π can never appear in BFS algorithm, we can consider that in an adjacency list of 1 2 precedes 5. In some case, we can dequeue 2 prior to 5, where 3 and 4 pi are equal to 2, however this won't work. We can apply the same logic by reversing 2 and 5, where 5 precedes 2 and 3 and 4 pi are now equal to 5, which also won't work. Hence we can conclude that a simple path in G' from the source vertex, 1, to any other vertex will be the shortest path in G.

For the next two problems, use the following graph $G = (V, E)$.



Problem 4 (20 points)**.** Describe the order in which nodes of the graph G are processed in BFS when the start node is 8 and neighboring nodes are taken in increasing order (smaller labels are enqueued first).

Solution. The order of nodes is 8,0,4,1,2,3,6,7,5 because, starting at the start node, 8, the node will be marked as visited, the adjacent nodes are left to explore, and are queued in ascending order. Then the search moves to the next queued node.

Problem 5 (20 points)**.** Describe the order in which nodes of the graph G are processed in DFS when the start node is 8 and neighboring nodes are taken in increasing order.

Solution. The order of nodes is 8,0,1,4,3,2,5,6,7 because, in depth-first search, it begins at the starting node and that node is marked as visited. It then processes the first adjacent node, which in this case it will follow an ascending order. If there are no unvisited adjacent nodes, the search must backtrack through the stack, and check all the conditions of each node until an unvisited adjacency appears.

Checklist:
☐ Did you add your name?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)

☐ Did you sign that you followed the Aggie honor code?
☐ Did you solve all problems?
☐ Did you typeset your homework in LaTeX and submit the resulting pdf file?