CSCI2300 – Introduction to Algorithms
Spring 2021, Exam II (100 Points + 5 Bonus)

Academic Integrity

This is an open book exam. You are only allowed to use a calculator for the computations. You must show all work for full credit.

Q0. (0 points) Read the following statements on academic integrity.

- I will not use any unauthorized materials (e.g., from the web, or other students).
- I will not use any unauthorized communication (e.g., messaging or other apps).
- I will not use any unauthorized means (e.g., any program/tool other than calculator).

Write the following statement at top of your answer sheet, followed by name and signature:

I attest that all work is my own and I have neither sought nor offered any unauthorized aid.

Name: ________________________ **Signature:** ________________________

Q1. (25 points) Consider the FFT approach to multiply two polynomials $A(x)$ and $B(x)$. Let the result of the evaluation at four points x_0, x_1, x_2, x_3 (powers of n-th root of unity) be given as follows:

$$[A(x_0), A(x_1), A(x_2), A(x_3)] = [2,\ 3+i,\quad 0,\ 3-i]$$
$$[(B(x_0), B(x_1), B(x_2), B(x_3)] = [3,\ 1+2i,\ -1,\ 1-2i]$$

Answer the following questions:

(a) (10 points) What is $C(x) = A(x) \cdot B(x)$ in value space?

(b) (5 points) Here $n = 4$. What is the value of n-th root of unity ω, and what is ω^{-1}?

(c) (10 points) What is the coefficient representation of $C(x)$? Note: To solve this, you may use $C(x) = \frac{1}{n} FFT\big([C(x_0), C(x_1), C(x_2), C(x_3)], \omega^{-1}\big)$. You can also use other approaches (show all work).

Q2. (25 points) Given an undirected, connected graph $G = (V, E)$, define *cvertex* to be a vertex $u \in V$ such that if we remove u and all its incident edges (u, v), the remaining graph is still connected. Answer the following questions:

(a) (10 points) Give a linear time algorithm to check if a given vertex u is a cvertex.

(b) (10 points) Give an algorithm to find *all* cvertices in G, and analyze its running time in terms of $|V|$ and $|E|$.

(c) (5 points) Give a linear time algorithm to find *all* cvertices in G.

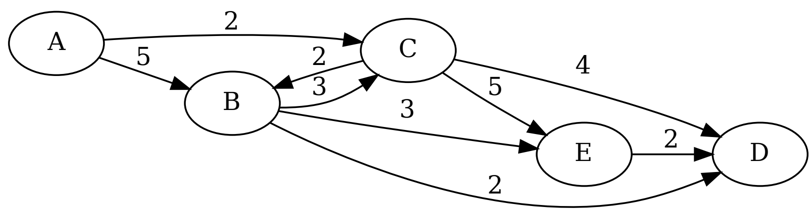

Figure 1: Graph for Q3

main(G^R, s):
for $u \in V$ **do**
  $visited[u] = 0$
  $D[u] = \infty$
end
$D[s] = 0$
for $u \in V$ **do**
  if *not visited[u]* **then**
    explore(G^R, u)
  end
end

explore(G^R, u):
$visited[u] = 1$
for $(u, v) \in E^R$ **do**
  if *not visited[v]* **then**
    explore(G^R, v)
  end
  if $D[u] > D[v] + w(u, v)$ **then**
    $D[u] = D[v] + w(u, v)$
  end
end

Algorithm 1: Shortest Paths on Reverse Graph via DFS

Q3. (25 points) Given a directed, weighted graph $G = (V, E)$, let $G^R = (V, E^R)$ be its reverse graph (obtained by reversing all edges). Consider algorithm 1 to find the shortest path from a given vertex s to all other vertices, by performing a DFS on G^R, and maintaining an array D, such that $D[u]$ keeps track of the shortest path length from s to v. Initially, the algorithm sets $D[s] = 0$. Then in **explore** it performs a DFS on G^R, and after each edge $(u, v) \in E^R$ is processed it updates the $D[u]$ value if a smaller value is found (note that $D[u]$ may be updated even if u has been visited before). Answer the following questions:

(a) (10 points) Apply algorithm 1 on the graph given in fig. 1. Show the edges that make up the shortest paths, and the distances.

(b) (10 points) Apply Dijkstra's algorithm on the graph in fig. 1. Show the edges that make up the shortest paths, and the distances.

(c) (5 points) Either prove that algorithm 1 is correct, or give a counter example to show it is not correct and determine on which class of graphs will it work correctly (if any)?

Q4. (25 points) Instead of union-find by rank, consider two variants that do union-find by *weight* and by *height*, both without path compression. Weight is defined as the number of nodes in the tree (stored at the root), and height is the height of the root node. The find operation remains unchanged, but in union you have to adjust either the weight or height (instead of rank). If you merge two roots with the same value, then choose the lower id node as the new root, otherwise the root with larger weight/height is the new root. Answer the following questions:

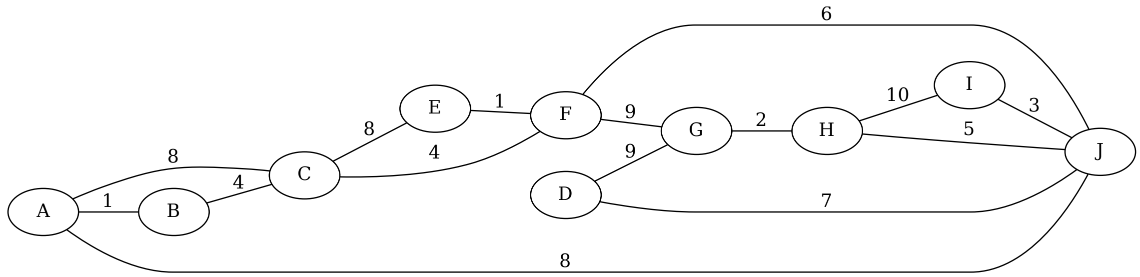

Figure 2: Graph for Q4

(a) (10 points) Given the weighted undirected graph in fig. 2, show the resulting trees for Kruskal's algorithm with union-find using both i) weight, and ii) height. Make sure to show the weight/height of each node in the tree. Note: For two edges with same weight, the alphabetically smaller one comes first.

(b) (10 points) Prove or disprove: The trees produced by using weight and using height are always the same (other than weight/height information).

(c) (5 points) What is the maximum height of any root (say, with weight k) when using union-find by weight?

Q5. (**Bonus**: 5 points): Match the following five answers with the correct person, in response to the question "why did the chicken cross the road?" Here are the statements:

1. I envision a world where all chickens will be free to cross roads without having their motives called into question.
2. In my day, we didn't ask why the chicken crossed the road. Someone told us that the chicken had crossed the road, and that was good enough for us.
3. Whether the chicken crossed the road or the road moved beneath the chicken depends upon your frame of mind.
4. The chicken did not cross the road. I repeat, the chicken did **not** cross the road.
5. Did the chicken cross the road? Did he cross it with a toad? Yes the chicken crossed the road, but why it crossed, I've not been told!

Here are the persons:

a. Richard M. Nixon

b. Martin Luther King, Jr.

c. Dr. Seuss

d. Grandpa

e. Albert Einstein