List of collaborators:

---

1. The algorithm discussed in class for computing pagerank includes a formula computing, for each vertex v, the value

$$P[v, i] = 0.05/n + \sum 0.95 P[u, i-1]/N[u]$$

where the sum is taken over all vertices u that have edges $u \to v$ and where $P[u, i]$ is a previously-computed quantity and $N[u]$ is the number of edges out of u. For some variations of the adjacency list representation, this formula can be computed efficiently, taking only constant time for each edge $u \to v$. However, a direct implementation of this formula using the Python graph representation described in https://www.python.org/doc/essays/graphs/ would not be as efficient. Explain why not.

∎

---

2. For CS 163: Describe the additional information that would be needed in an adjacency list representation to make the computation of these sums efficient.

For CS265: Describe a method for computing the same values $P[v, i]$, for all vertices v, that uses only the Python graph representation and takes constant time per edge. (Hint: for each vertex, instead of finding the value of its own sum, compute its contribution to the sums of all the other vertices.)

∎

---

3. (a) Write down the Python representation (as described in the link for problem 1) for a directed graph whose vertices are the integers from 0 to 7, where each vertex numbered i has outgoing edges to the vertices numbered $i+1$ (mod 8) and $2i$ (mod 8). (If these two numbers are equal, there should be only one outgoing edge. For this problem, it is allowed for a vertex to have an edge to itself. If you are using this template to format your answers in LaTeX, you need to put a backslash character in front of each curly bracket, like \{. . .\}, to make them visible.)

(b) In what order would these vertices be visited by a depth-first search, starting from vertex 0? When the depth-first search algorithm loops over the neighbors of a vertex, it should do so in the order given in part (a).

(c) In what order would these vertices be visited by a breadth-first search, starting from vertex 0?

∎

---

4. Write an adjacency matrix representation for the graph from problem 3.

∎