

Introduction

This pre-lab is designed to help prepare you to complete the programming component of M11's Lab.

Because the pre-lab section is designed to help you plan your implementation, it is strongly recommended for you to complete this section **before** writing any code. However, it is also natural for your understanding of the implementation requirements and your own strategy to evolve as you complete the code, so feel free to update your answers to the pre-lab section during/after the coding phase of the lab, up to the pre-lab deadline.

Note: unlike previous pre-labs, this pre-lab writeup will be graded for correctness of answers given rather than sufficient effort alone.

Questions

1. Given a **Vertex** object x , give Java code to enumerate its outgoing edges.
2. Given a **VertexAndDist** object x and a new distance d , give Java code to create an updated object with the same vertex as x but with distance d .
3. For simplicity of implementation, we use HashMaps to map vertices to handles/parents and edges to weights. How could you modify the **Vertex** and/or **Edge** classes, as well as the maps themselves, to implement the maps using ordinary arrays, with no hashing? Be sure to address both the vertex and edge maps in your answer. (*Hint*: consider the **Vertex**'s "id" field for inspiration.)

For questions 4-7, please be sure to **justify** the running times you claim using what you know about the cost of Dijkstra's algorithm and the meaning of dense and sparse graphs.

4. Suppose we know that our input graph $G = (V, E)$ is dense. What is the asymptotic running time of Dijkstra's algorithm on G in terms of the number of vertices $|V|$?
5. Now suppose we know that our input graph $G = (V, E)$ is sparse. What is the asymptotic running time in terms of $|V|$?

A *Fibonacci heap* is a fancy priority queue data structure. For a heap of size n , it takes $\Theta(\log n)$ time to do an *extractMin()* operation but only $O(1)$ time to do an insert or decrease operation. Suppose we replace the binary heap used in Dijkstra's algorithm by a Fibonacci heap.

6. If the graph is dense, what is the asymptotic complexity of Dijkstra's algorithm using a Fibonacci heap, in terms of $|V|$?
7. If the graph is sparse, what is the asymptotic complexity of Dijkstra's algorithm using a Fibonacci heap, in terms of $|V|$?