Currently, I'm trying to pass in a vertex struct to my priority queue, and the comparator orders the heap by vertex weight. (Weight is the cost of the edge + weight of the previous node).

When I do an extractBinomial, I run through the adjacency list of the vertex, but I don't know where to store the new weights that gives me. I think I should put a new vertex with that weight in my heap.

I also don't know what the update function that we pass into our heap does. The BinomialNode value is private, so I don't know how to make update access it, or what update should even do.

The update function should update the binomial node pointer within your vertex when you bubble up your vertex values, since they need to know which binomial node they belong to.

I am not sure what you mean by "where to store the new weights that gives me". You should extract the vertex with the minimum distance. Then for all the neighbors update their distance and then if needed update their distance in the heap.

The update function takes in the Binomial Heap the value is stored in and a void pointer to the value you store in the Binomial Heap.

For this assignment the value is a pointer to a vertex object. This means that your update function knows all about what the value actually represents and you can cast the void pointer in your update function to a vertex and update the Binomial Node appropriately.