Subject: Vertex Class and its contents  
Posted by nltollman on Fri, 18 Nov 2016 04:41:16 GMT

I discuss a topic and understand what I am to do but then hours later I come back to it and lose track yet again, even when I made what seemed like amazing notes at that time. Anyways! Figured I would make a topic that I can refer back to if this happens again.

I definitely know that reading the file the first time deals with the vertex class, by getting the largest vertex and largest weight.

1. The vertex class is where not only the vertexComparator function will be held for the binomial heap to use, but also where you will be making the AdjMatrix of weights and AdjList of vertex objects, correct?

If that is the case, then there are 3 different structs? one for the vertex object, one for the matrix, and one for the list?

2. The AdjMatrix is a temporary storage unit to place the weights that will be used by what later? I forget... (I know that the dimension is the largest vertex + 1)

3. The AdjList is where all of the vertex objects being made from the vertex, edge, and weight being read in are stored and used later by Prim's? (I know that Prim's "infinity" is the largest weight + 1)

4. Does the binomial heap use one of these to help create the heap? I have most of the psuedocode created for the binomial class from Lusth's sticky he has up. (I still need to do the bubble up in decreaseKey, and complete combine and consolidation)

Basically....I need help remembering the use of the AdjMatrix and AdjList after they are created.

Also, as a note, I will be in Rodgers all day tomorrow working on this and other classwork, if anyone is interested. Thanks in advance! :)

Subject: Re: Vertex Class and its contents  
Posted by lusth on Fri, 18 Nov 2016 11:55:38 GMT

I'm not going to use an AdjList; you can get the connections from the AdjMatrix. I will also have an array of vertex objects, the vertex 0 object will be stored at index 0 in that array, and so on.

The binomial heap is independent of the graph information, except for bubbleUp, which needs to inform a value that it's owner has changed (see the binomial heap pseusocode thread).

Prim's algorithm uses the AdjMatrix to find adjacent vertices and the weights of those connecting edges.
Is the owner of a vertex a pointer to the vertex that it is adjacent to? So, if you have 1 5 1; and 1 7 23;, the owner of vertex 5 with weight 1 and vertex 7 with weight 23 would be vertex 1?

5(1)1, 7(1)23;
weight : 24;
where 1 is the owner of 5 and 7 and the total weight is 24?

If that is the case, then if you have 1 5 1; 1 7 23; and 7 5 4;, then
5(1)1;
7(5)4;
weight : 5
where 1 is the owner of 5 and 5 is the owner of 7.
And since 5 is the owner of 7, then the total weight changes from 24 (if 1 was the owner of both) to 5?

Would the owner be a vertex pointer or a node pointer?

The owner of a vertex is the node in the binomial heap that holds the vertex as its value. Once you have created a vertex object, you set its owner by inserting it into the binomial heap:

v->owner = insertBinHeap(b,v); //v is a vertex, b is a binomial heap

The insert routine for a binomial heap should return the node that encapsulates the vertex.

so to keep this attachment of vertices and owner all assignments that need it is when call insert by assigning v->owner to the return and when call decrease key by assigning the new value to the returned node?. Also can any one explain the predecessor roll so can have an idea on how to initialize it ?.