Subject: Accessing BinomialNode within the Vertex to pass to Decrease-Key
Posted by kamadson on Mon, 27 Mar 2017 21:59:40 GMT

View Forum Message <> Reply to Message

In my deleteBinomial function, I pass in a BinomialNode *n as a parameter like the pseudocode. I know that the node we are passing to decrease-key is supposed to be the BinomialNode *holder that is within the Vertex object, but in order to access that node, don't you have to say specifically, Vertex *v = n->value, and then pass v->holder in my call to decrease-key?? Or is there a way to not directly reference my Vertex object? I just thought these programs were supposed to be generic, so I feel like I should not be using my Vertex pointers at all in bin heap.c. Hopefully this makes sense..

Subject: Re: Accessing BinomialNode within the Vertex to pass to Decrease-Key
Posted by jarobinson3 on Mon, 27 Mar 2017 22:09:17 GMT

View Forum Message <> Reply to Message

I think in most students implementations the value saved in the BinomialHeap as the vertex. Then you use the update/compare operators to compare two values and also to update the BinomialNode stored in each vertex.

Honestly, I am not really sure what the point of decreaseKeyBinomial is other than to notify the BinomialHeap that the key changed and we should update it. I guess we should have called it notifyKeyChangeBinomial(Binomial*,BinomialNode*);

Subject: Re: Accessing BinomialNode within the Vertex to pass to Decrease-Key
Posted by SSinischo on Mon, 27 Mar 2017 23:24:45 GMT

View Forum Message <> Reply to Message

jarobinson3 wrote on Mon, 27 March 2017 17:09
Honestly, I am not really sure what the point of decreaseKeyBinomial is other than to notify the BinomialHeap that the key changed and we should update it. I guess we should have called it notifyKeyChangeBinomial(Binomial*,BinomialNode*);

decreaseKeyBinomial actually does the job of changing the values in most applications. (like if you were to use integers with your heap) It is just not seen in this implementation since the pointer to the value it holds stays the same.

Subject: Re: Accessing BinomialNode within the Vertex to pass to Decrease-Key
Posted by lusth on Tue, 28 Mar 2017 12:10:33 GMT

View Forum Message <> Reply to Message

kamadson wrote on Mon, 27 March 2017 16:59:In my deleteBinomial function, I pass in a
BinomialNode *n as a parameter like the pseudocode. I know that the node we are passing to
decrease-key is supposed to be the BinomialNode *holder that is within the Vertex object, but in
order to access that node, don't you have to say specifically, Vertex *v = n->value, and then pass
v->holder in my call to decrease-key?? Or is there a way to not directly reference my Vertex
object? I just thought these programs were supposed to be generic, so I feel like I should not be
using my Vertex pointers at all in bin heap.c. Hopefully this makes sense..

Your binomial heap should not reference vertex objects.

My dijkstra application does, though. It calls decreaseKey, passing the holder of the vertex (a
binomial node). I use the vertex itself as the new key (the comparator wants vertices as keys to
compare).

Subject: Re: Accessing BinomialNode within the Vertex to pass to Decrease-Key
Posted by jarobinson3 on Tue, 28 Mar 2017 14:19:20 GMT
View Forum Message <> Reply to Message

SSinischo wrote on Mon, 27 March 2017 18:24
decreaseKeyBinomial actually does the job of changing the values in most applications. (like if
you were to use integers with your heap) It is just not seen in this implementation since the
pointer to the value it holds stays the same.

Only if the key and value are the same and you have no access to the original pointer would
assignment make sense. In all other cases you just change the key externally and do an
assignment that does nothing then the tree gets updated.