Subject: comparator again
Posted by oamohamed@crimson.ua.edu on Wed, 16 Nov 2016 03:32:51 GMT

my binHeap has an attribute: int(*comparator)(void *,void *); on a vertix class i have this method that it seems like a very valuable func if I understood it correctly;
int lessthann(void *a,void *b) "which it return the a- b value" and it reside in a vertex class. my question is this; will my initialization be right if I did something like this on consildate(b) or evenon bubbleup for example.
on consolidate(binHeap *b){
    b=newBinHeap(*lessthann);//will this initialization be valid?.
}

Subject: Re: comparator again
Posted by maswain1 on Wed, 16 Nov 2016 03:49:02 GMT

If I understand correctly, once you've defined the comparator function in your vertex class and include "vertex.h", you can declare a binomial heap like this: binHeap* b = newBinHeap(vertexComparator) or something similar. So far, I've been doing it this way and my binHeap class seems to be referencing/using the function just fine.

Subject: Re: comparator again
Posted by lusth on Wed, 16 Nov 2016 14:37:14 GMT

b=newBinHeap(*lessthann);//will this initialization be valid?.
No, you need to pass a pointer to the function. The asterisk in front of lessthan is derefrencing the pointer.

Subject: Re: comparator again
Posted by berryhilleric on Tue, 29 Nov 2016 16:42:35 GMT

[NEVERMIND, FOUND THE ANSWER IN THE SECOND PAGE OF THE "TEST CASES" THREAD]

Just so I'm completely clear, the priority when comparing two vertices w/ the comparator is:

1. Compare their keys. If they're equal, then
2. Compare their vertex values. If they're equal, then
3. Compare their predecessors values.
Its this order that gets me the output that matches the output from the test cases provided by btlindow. In class we were told that the predecessor value should come before the vertex value, which gives me a completely different shortest path.