Edit 1: Aannd apparently this is probably wrong, cuz apparently someone did pass test #17 and they said they didn't call update in insert, so now I'm back to knowing nothin.

Original post:
I'm pretty sure I figured out what caused an error in Test #17- although the pseudocode does not specify to do this, update should be called in insertBinomial in order to initialize the binomialNode pointer that the vertex will have.

You, like me, may not have thought to do this, because in your dijkstra.c you initialize the vertex's pointer to the result of insertBinomial. I thought the whole reason why insertBinomial returns a pointer was for this very reason. I didn't consider that that functionality should happen within binomial.c.

So without that functionality, a call to decreaseKey in Lusth's dijkstra on a binomialNode which holds a vertex which has never had its pointer initialized will result in an error.

Subject: Re: CALL UPDATE IN INSERT
Posted by SSinischo on Sun, 02 Apr 2017 22:47:11 GMT

How are you sure this is what caused your failure?

Calling update in insertBinomial would be redundant. That's the whole reason the function returns a BinomialNode.

Subject: Re: CALL UPDATE IN INSERT
Posted by davidmccoy on Sun, 02 Apr 2017 22:57:04 GMT

Hans said he did have update in insert and he passed test #17, and I can't think of anything else I'm doing wrong, so it's a guess for now, but I think it's a good guess.