Edit 2: Looks like you never use darrayPointer[index], because we're obviously using the get and set functions to reference places in arrays. So that's the solution for their use in binomials. duh (:  
Edit 1: Looks like there are several points in the binomial heap functions which need to go through the DArray. Guess we can just peruse it from binomial.c in a loop or whatever.

//remove y from b's root list (place a null in y's spot)

//step 1: I need to get index of y

//step 2: update y to be null
setDArray(b->rootlist,index,NULL);

So for extractBinomial, in the class pseudocode, it says we need to remove y from b's root list. We have a DArray function that can update a value at a given position, but to update y, we need to know where it is in the array. There is no function provided by DArray to get the index of a value, so we would need to find y by manually searching through the rootlist. Is it legal for us to search from 0 to sizeDArray to find y's position using the compare function, and then call setDArray? That seems to break the rule that binomial.c should know nothing of the implementation of DArray.h.

---

Subject: Re: extractBinomial: Removing y from b's root list
Posted by jarobinson3 on Sat, 01 Apr 2017 16:00:46 GMT

View Forum Message <> Reply to Message

Just iterate through DArray. It has a function for size and getting a value at a position.