CS3130, Spring 2017, Dr. W. He

Project 3: Apply Heapsort on Dynamic Data

(Due May 7/14, 2017)

Description:

In this project, we assume that our data comes in *dynamically*. We need to maintain our data in a heap, after every *insertion* and *deletion*. We also need to handle the *underlying* array dynamically. Whenever we detect an *overflow* in our array, we will create a new array with size doubling the previous size.

Requirements:

1. (*Dynamic Data*) When we generate the data, we *simulate* dynamic data. We use a 2000-element integer array containing 2000 random integers with values between 1 and 2000 inclusive. But we do not generate them at the same time. Every time we generate 100 random integers.

2. (*Initial array*) When our data grows, we need to change the array size if an overflow is detected. Initially, create an array with size 100 to hold the first 100 random integers. Then build a heap immediately.

3. (*Copy data*) Before you insert a new integer into the array, check if there would be an *overflow* if the number is inserted. If yes, create a new array with size doubling the previous size and copy data from the old array to the new array, then insert the new integer.

4. (*Recover heap*) After every data insertion, fix the heap immediately.

5. (*Heapsort*) At the end when all the data comes in, apply the heapsort on the heap.

6. (*Display information*) Every time when you generate 100 random integers, print them out on the screen with 20 numbers per line. Every time when you double the array size, print out the sizes before and after the change. At the end, print out the sorted array.