Do NP-completeness homework, Part 4.

1. Consider an array of size nine with the numbers in the following order

$$40, 10, 20, 30, 90, 80, 70, 60, 50.$$

   (a) Form the heap. Show the heap as a tree. Show the heap as an array. Exactly how many comparisons did heap creation use?

   (b) Start with the heap created in Part (a). Show the *array* after each element sifts down *after heap creation*. How many comparisons does each sift use? What is the total number of comparisons *excluding heap creation*?

2. We are going to derive a good upper bound on the worst case number of comparisons in heapsort. In order to simplify the algebra assume that the size of the list that you are sorting is one less than a power of two (i.e., a full tree). Thus, an array of size $n = 2^k - 1$ is associated with a binary tree with k levels numbered $0, 1, \ldots, k-1$.

   (a) How many nodes are on level j?

   (b) How many comparisons does the heap formation phase use to sift an element rooted on level j in the worst case.

   (c) Give a summation for the total number of comparisons for heap formation.

   (d) Simplify the summation. Show your work.

   (e) How many comparisons does the remainder of heapsort use for each sift after removing an element from level j? Note that the tree gets smaller so not all elements on level j have exactly the same number of comparisons.

   (f) Give a summation for the total number of comparisons for the second phase of heapsort.

   (g) Simplify the summation. Show your work.

   (h) Add the two totals and simplify.