Problem 1. Consider an array of size eight with the numbers in the following order $40, 20, 80, 60, 30, 10, 70, 50$.

(a) What is the array after heap creation? Make sure to form the heap bottom up as done in class. How many comparisons does the algorithm use?

(b) Show the array after each element sifts down during the remainder of heapsort, and state how many comparisons *each* sift takes. What is the total number of comparisons for the remainder of heapsort (i.e., the sum of the comparisons for all of the sifts)?

Problem 2. We are going to investigate the spatial locality of Bubble Sort. For concreteness, the computer has one local *page* of size s, and a small smount of extra memory that is also local. At any given time the local page can hold (the s) array elements indexed by $1+is,\ 2+is,\ 3+is, \ldots,\ s+is$ (for $i = 0, 1, 2, \ldots$). The cost of bringing a page into local memory is β. We simplify this by charging β for the first comparison to an element within the page. After that, comparisons to elements in the local page only cost α (where $\alpha < \beta$). The elements previously in the page become nonlocal. To keep the asymptotic computations simple, assume that s is asymptotically larger than β/α.

For example, assume that $s = 4$ (and n is much larger than 4). The first comparison between A[1] and A[2] is nonlocal and costs β, as A[1], A[2], A[3], and A[4] are brought into the local page. The next two comparisons between A[2] and A[3] and between A[3] and A[4] are local and each costs α. The next comparison between A[4] and A[5] is nonlocal and costs β, as A[5], A[6], A[7], and A[8] are brought into the local page (and A[1], A[2], A[3], and A[4] removed from the local page). Etc.

(a) What is the cost of Bubble Sort (measuring just the comparisons)? Get the exact high order term assuming that n is asymptotically larger than s. Show your work and justify.

(b) If all of memory were one page then the asymptotic, spatial locality (for comparisons) would be $(n^2/2)\alpha$. The ratio of this value and the value from Part (a) is the *asymptotic locality*. What is the asymptotic locality of Bubble Sort?

Problem 3. We are going to investigate the spatial locality of Heap Sort. We will use the same assumptions as above.

(a) What is the worst-case cost of Heap Sort (for comparisons)? You can assume that s is a power of 2. Get the exact high order term assuming that $\log n$ is asymptotically larger than $\log s$. Show your work and justify.

(b) If all of memory were one page then the asymptotic, spatial locality (for comparisons) would be $2\alpha n \lg n$. The ratio of this value and the value from Part (a) is the *asymptotic locality*. What is the asymptotic locality of Heap Sort?