

Due at the start of class Wednesday, June 17, 2015.

Problem 1. For this problem (as always) we will use Williams's original heapsort algorithm with Floyd's modification for heap creation. Consider an array of size eight with the numbers in the following order 10, 30, 50, 70, 20, 40, 60, 80.

- (a) What is the array after heap creation? How many comparisons does heap creation use?
- (b) Starting from the heap, show the array after each sift operation and state how many comparisons each sift takes. How many comparisons does the algorithm use in total after heap creation?

Problem 2. We are going to find the exact number of comparisons (in the worst case) for heap creation in a complete binary tree, where the last level is completely full. (So if the last level of the tree is k it has exactly 2^k nodes.) We will guess the formula by looking at a few small examples, and then prove it is correct by mathematical induction.

- (a) Calculate by hand the exact number of comparisons for complete trees with 0, 1, 2, 3, 4 levels.
- (b) We know that the true answer should be approximately $2n$. Find the differences between $2n$ and your calculated values.
- (c) Guess a formula for your differences as a function of n .
- (d) What formula does that give you for the exact number of comparisons for heap creation as a function of n ?
- (e) Heap creation can be thought of as a recursive procedure: Create heap for left child of root, create heap for right child of root, and sift down root value. Write a recurrence for the number of comparisons to create a heap.
- (f) Use mathematical induction to prove that your formula is a solution to the recurrence.