Due date: Email your solutions to cs140_17fall@163.com or submit a written copy to a TA before 7:30 PM, October 19.

Problem 1 Given an array with 14 values $[1, 8, 15, 17, 18, 12, 20, 11, 26, 6, 9, 5, 35, 13]$, build a max binary heap from the array. Draw the heap you end up with.

Problem 2 Median-heap. Suppose you are given a stream of values and want to maintain the median at any point in time. For example, if the stream is 4,1,5,6,7,2,..., then your algorithm should output 4 after seeing the first input value, output 1 after seeing two values (for an even number of inputs, choose the smaller of the two middle values as the median), output 4 after 3 inputs, output 4 after four inputs, etc. Solve this problem using only two heaps and a constant number of auxiliary variables. Your algorithm should perform $O(n \log(n))$ steps after seeing n input values.

Problem 3 Young tableaux. An $m \times n$ Young tableau is an $m \times n$ matrix such that the entries in each row are in sorted increasing order from left to right and the entries of each column are in sorted order from top to bottom. Some of the entries of a Young tableau may be ∞, which we treat as nonexistent elements. Thus, an $m \times n$ Young tableau can be used to hold up to mn finite numbers.

(a) Draw a 4×4 Young tableau containing the elements $\{47, 54, 5, 45, 98, 14, 11, 24\}$.

(b) Argue that an $m \times n$ Young tableau Y is empty if $Y[1, 1] = \infty$. Argue that Y is full (contains mn elements) if $Y[m, n] < \infty$.

(c) Give an algorithm to extract the minimum element of the tableau in $O(m+n)$ time. Afterwards, the matrix should still be a Young tableau.

(d) Show how to insert a new element into a nonfull $m \times n$ Young tableau in $O(m + n)$ time.

(e) Show how to use an $n \times n$ Young tableau to sort n^2 numbers in $O(n^3)$ time. Do not use any other sorting methods as subroutines.

(f) Give an $O(m + n)$-time algorithm to determine whether a given number is stored in an $m \times n$ Young tableau.