Due date: Email your solutions to cs140_17fall@163.com or submit a written copy to the TA before 7:30 PM, November 16, 2017.

Problem 1 Given two sorted arrays of sizes m and n respectively, describe an algorithm to find the median of the values from both arrays, which runs in $O(\log(m+n))$ time.

Problem 2 Given a convex function $f(x), x \in [a, b]$, find the minimum value of f. You are not given an explicit expression for f, but can evaluate $f(x)$ at any $x \in [a, b]$.

Problem 3 Consider a knapsack with capacity W which you want to pack with different materials. There are N materials, and the i'th material has weight w_i and value v_i. Your goal is to select an amount $0 \leq x_i \leq w_i$ of each material (x_i can be fractional), that maximizes the total value $\sum_{i=1}^{N} x_i v_i$ in the knapsack, subject to the capacity constraint $\sum_{i=1}^{N} x_i \leq W$. Give an $O(n \log(n))$ time algorithm for this problem and prove that it is correct.

Problem 4 Describe an efficient algorithm that, given a set of points $\{x_1, x_2, ..., x_n\}$ on the real line, computes a minimum cardinality set of unit length closed intervals whose union contains all the points. Argue that your algorithm is correct.

Problem 5 Given an $m \times n$ matrix of numbers A, we say an element is a *peak* if it is not smaller than its neighbors. I.e. for $0 \leq i < m$ and $0 \leq j < n$, $A_{i,j}$ is a peak if $A_{i,j} \geq \max\{A_{i-1,j}, A_{i+1,j}, A_{i,j-1}, A_{i,j+1}\}$. We set all boundary values outside the matrix to be $-\infty$, i.e. we set $A[i,-1] = A[i,n] = -\infty$ for $0 \leq i < m$, and $A[-1,j] = A[m,j] = -\infty$ for $0 \leq j < n$.

1. Circle all peaks in the matrix below.

$$\begin{matrix} 20 & 9 & 5 & 5 & 9 \\ 5 & 10 & 10 & 8 & 4 \\ 3 & 3 & 1 & 8 & 6 \\ 10 & 30 & 15 & 45 & 7 \\ 6 & 10 & 20 & 30 & 40 \end{matrix}$$

2. Prove that any matrix A has at least one peak, and give an $O(m \log(n))$ time algorithm to find a peak.

3. Consider the following algorithm.

   - Treat the middle column as an $m \times 1$ matrix, and find a peak at some index (i, j).
   - Consider the two values $A_{i,j-1}$ and $A_{i,j+1}$. If neither value is greater than $A_{i,j}$, then return $A_{i,j}$ as a peak. Otherwise, recurse on half of the matrix containing a greater value.

   (a) Analyze the time complexity of this algorithm.

   (b) Is this algorithm correct? If so, argue why. If not, provide a counterexample.