Important: Be clear and precise.

We recommend that you do these problems on your own. Do not look for answers elsewhere and do not discuss these with anybody. In case you already know the answer mention this in your submission. If you do discuss the problem with somebody by accident or otherwise mention this too in your submission. You get half the marks for attempting most questions, so attempt each one.

You get points for what your TA makes of what you have written. Make reasonable assumptions when necessary, though we reserve the right to determine if they were indeed reasonable. If you write many answers to the same question, only the first will be graded. You can use any theorem/algorithm done in class. You have to describe the algorithm in English. You may then give pseudocode. However there must be enough detail in the description so that if we wish to write code we can. Writing code is optional, but writing only code will not fetch you any marks. Clearly indicate the recurrence and the table entries in case of DP. Submit these on A4 sized sheets: HW2-A. In case running times are not explicitly mentioned in the question, any polynomial in the input size is all right.

1. Given an array $A[1, \ldots, n]$ of integers, design an algorithm to compute an array $X[1, \ldots, n]$ such that $X[j]$ contains the smallest integer in A that ends with an increasing sequence of length j. This procedure should run in $O(n \log n)$ time.

   Hint. X has a nice property. If you cannot get it, see a few examples.

2. You are given two sequences of symbols from an alphabet. Say $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. The output we desire is the length of the longest sub-sequence that is common to both. Design an algorithm for this problem.

   For instance consider $2, 4, 3, 2, 7, 2, 1, 8, 1, 7$ and $5, 1, 3, 2, 1, 7, 2, 8, 3, 4$. The answer is 5. The subsequence is $3, 2, 7, 2, 8$

3. Given a set of intervals S with weights design an algorithm to find a minimum weight subset T such that every point covered by an interval in S is also covered by some interval in T.

4. Given a sequence of weighted points in the positive quadrant design an algorithm to find the maximum weight sequence of points so that each successive point is contained in the rectangle formed by the previous point and the origin.