Data and File Structures 2018
(Problem Set 1)

August 13, 2018

- Answers to all problems are not to be submitted. The ones whose solutions are to be submitted is marked.
- Due on Wednesday, August 24, in the class.
- Late home works will not be accepted.
- Please give precise arguments for all statements that you write.
- If you are given to write an algorithm or a procedure, writing the pseudocode is not enough. Please give an informal explanation of the procedure in words.
- Please do not hesitate to contact me if you do not understand the problems.
- Collaboration is discouraged, but not prohibited. It is recommended that you try to solve the problems on your own. You can discuss the questions with your colleagues but you should not copy solutions. Always write down your own answers. **Copying is a serious offence and is dealt with very strictly**.
- Credits would be given to partial solutions also.
- The answers should be typed or written clearly and a hard copy is to be submitted.

1. [**To be submitted**] Give an algorithm to find both the maximum and minimum of an array of n numbers. Try to make it as efficient as possible. (Hint: The best algorithm would only use around $\frac{3}{2}n$ comparisons).

2. Consider the problem of evaluating a polynomial at a point. Given n integer coefficients $a_0, a_1, \ldots, a_{n-1}$ and and integer x, we want to compute $\sum_{i=0}^{n-1} a_i x^i$. Give an algorithm to do this. How many additions and multiplications are required to do this. Try to

make your algorithm as efficient as possible. (Hint: The best algorithm would use about n multiplications and n additions)

3. Let A be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A.

   (a) List the inversions of the array $< 2, 3, 8, 6, 1 >$.

   (b) Which array containing the elements from the set $S = \{1, 2, 3, \ldots, n\}$ contains the maximum number of inversions. How many does it have?

   (c) What is the relationship between the running time of *insertion sort* and the number of inversions present in the array being sorted? Justify your answer.

4. **[To be submitted]** Prove the following:

   (a) $\log(n!) = \Theta(n \log n)$

   (b) $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$

5. Write a routine to print a link list in reverse order using only $O(1)$ extra space. Note: Because of the requirement of constant amount of extra space, recursion is ruled out.

6. A linked list contains a cycle if, starting from some node p, following sufficient number of links brings us back to node p. p is not necessarily the first node in the list. Assume you are given a linked list that contains N nodes; however the value of N is unknown.

   - Design an $O(N)$ algorithm to determine if the list contains a cycle. You may use $O(N)$ extra space.
   - Design an $O(N)$ algorithm to determine if the list contains a cycle using $O(1)$ space.

7. **[To be submitted]** We call a matrix a sparse matrix if it's entries contains a significant number of zeros. We store a $m \times n$ sparse matrix A in m interconnected link lists as described in Figure 3 (page 16) of David Mount's notes. Give a complete description of the above mentioned data structure to store a linked list. Give an efficient algorithm to multiply two sparse matrices A and B using the data structure.

8. **[To be submitted]** Propose a data structure that supports the stack Push and Pop operations and a third operation FindMin which returns the smallest element in the data structure, all in $O(1)$ worst case time and $O(1)$ extra space.