Data Structures (IT205)
Second Midsemester-semester Exam

Time: 2 hours **marks:4X15=60**

Answer any 2 from 1,2,3; Question 4 is compulsory; Answer one of questions 5,6

1. Consider the problem where you are given an array $A[1 \dots n]$ of n distinct integers. You are told that EXCEPT for positions $2^0, 2^1, \dots, 2^i$, where $i = \lfloor \log_2 n \rfloor$, the other elements of the input array are in their correct position in the ascending sorted order. Thus, for example elements in $A[3]$, $A[5]$ and $A[6]$ are the 3^{rd}, 5^{th} and 6^{th} smallest elements in the array respectively. However, no such guarantee is given about elements $A[1]$, $A[2]$, $A[4]$, $A[8]$ etc.

   Given this special information design an asymptotically optimal algorithm for sorting arrays of this type. Analyse the running time of your algorithm.

2. You are given an array $A[1 \dots n]$ of n distinct integers and also have the following extra information. The sub-arrays
   $A[1 \dots \lfloor \frac{n}{2} \rfloor]$
   $A[\lfloor \frac{n}{2} \rfloor + 1 \dots \lfloor \frac{3n}{4} \rfloor]$
   $A[\lfloor \frac{3n}{4} \rfloor + 1 \dots \lfloor \frac{7n}{8} \rfloor]$
   .
   .
   are all sorted subarrays, but no relation is known between elements in different subarrays.

   Using this extra information design an asymptotically optimal algorithm for sorting this type of array. Analyse the running time of your algorithm.

3. You have available an implementation of Quick-Sort which runs in time $15n \log n$ and an implementation Merge-Sort which runs in time $10 \log n$. If you need to sort an array of numbers occupying 800 MB of memory and the total memory available to you (including temporary memory) is 1200 MB, then which of the two implementations would you use? Explain your answer.

4. Consider the binary minimum heap represented by the following array.

1	17	35	77	91	49	47	85	105	99	103	67	70	56	60

   If we change the key value at $A[i]$ from a_i to b_i then the change c_i is defined as $|a_i - b_i|$. Assume we want to violate the minimum heap property, currently present in the heap A by changing exactly one key. Choose a position in the array such that the change c_i is minimum possible the resulting heap A' violates the minimum heap property. State the array position, whether we should increase the key value or decrease it and by how much.

5. What is the minimum possible **height** (NOT Black-height) and the maximum possible height of red-black trees containing 7 key values? Use the key values $\{1, \dots, 7\}$ and draw an example red-black tree T_1 with these keys of the smallest possible height and another T_2 of the largest possible height.

6. Consider the "red-black" tree specified by the following nodes (with attributes). The tree has only 5 key bearing nodes, and the attributes of the nil nodes have not been specified.

   - Node A: $parent = nil$ (ROOT); $left = B$; $right = C$; $colour = black$; $key = 75$
   - Node B: $parent = A$; $left = D$; $right = E$; $colour = red$; $key = 37$
   - Node C: $parent = A$; $left = nil$; $right = nil$; $colour = red$; $key = 85$
   - Node D: $parent = B$; $left = B$; $right = C$; $colour = black$; $key = 12$
   - Node E: $parent = B$; $left = B$; $right = C$; $colour = black$; $key = 44$

   Draw the above red-black tree along with its nil nodes. State the property of red-black trees which is violated by this purported red-black tree. Write code using **recolouring** of some nodes and/or **rotates** at a suitable node to restore the red-black trees. You should use as few operations of recolouring and rotate as possible. (Using a longer sequence than the minimum one will get less marks.)