PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. For all problems, use only the algorithms discussed in class/text.

2. Write your answers directly on the exam question sheet. You may use scrap paper (supplied by your proctor) for work, but these will not be graded.

3. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality.

4. You may use only a pen or a pencil. No calculators allowed.

Note. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality. The points assigned to each question are provided in parentheses.
1. (12) Consider a data structure with the operations: \textit{put}(x), \textit{find}(x), and \textit{remove}(x). \textit{Put}(x) inserts an element \(x\) into the structure, \textit{find}(x) searches for \(x\), and \textit{remove}(x) removes \(x\) from the structure. A \textit{find}(x) and a \textit{remove}(x) operation take 1 unit of time each and a \textit{put}(x) operation takes 1 unit of time except when the number of elements in the structure prior to the \textit{put}(x) operation equals the capacity \(n\) of the structure. In this case, \textit{put}(x) takes \(2n+1\) time units to double the capacity and then put \(x\) into the data structure. Assume that the structure is initially empty and its initial capacity is 1. Show that the amortized complexity of \textit{put}(x),\textit{ find}(x) and \textit{remove}(x) are respectively, 5, 1, and 1.
2. You are given $n=1000$ records to be sorted on a computer with a memory capacity of $S=100$ records. Assume that the entire $S$-record capacity may be used for input/output buffers, i.e. you have extra memory for a $k$-way loser tree. The input is on disk and consists of $m$ runs.

Assume that you use $2k$ buffers for input and 2 for output. Also assume that each time a disk access is made, the seek time is $t_s=10ms$ and the latency time is $t_l=5ms$. The transmission time is $t_t=0.5ms$ per record transmitted.

(a) What is the buffer size, $b$, and the total input time for phase two of external sorting, merging, if a $k$-way merge scheme is used? What is the better $k$ with respect to the total input time for phase two? Consider only $k=4$ and 9.

(b) Assume that it takes 0.5 ms to merge 10 records. Compute the total time for the phase two of external sorting, taken when $k=9$, i.e. compute the total time taken for input, merging and output.

(c) We can use either a winner tree or loser tree for external sort. Which will run faster in practice? Why?
3. (6+6)

(a) Convert the following *min tree* to a *height-biased min leftist tree* and label each node \( x \) with its shortest(\( x \)) value. Do this by swapping left and right subtrees as needed.

```
    1
   / \  
  4   5
 / \  
7   8
 \ / \  
 9 41 30
```

(b) Draw the min leftist tree that results when the *combine* operation is performed on the following two *min leftist trees*. (Use the algorithm discussed in the lecture.)

```
    1
   / \  
  4 6  
 / \  
8 9 10 11
/  
15
    3
   / \  
  7 5
 / \  
13
```
4. (9+3) a) Perform the following operations on an initially empty *min binomial heap* (showing each step).

   Insert(10), Insert(3), Insert(7), Insert(5), RemoveMin, RemoveMin, Insert(6), Insert(20), RemoveMin, Insert(3), Insert(4).

b) Give the **actual** complexities of *Insert, RemoveMin*(or RemoveMax) and *Meld* operations in a *binomial heap*. (No need to prove the complexities).