Small Group Tutorial 3, 20 - 24 March 2017

Please note that all tasks marked as "ADDITIONAL" will be done during the SGT session only if the rest of the exercises is done. Please try to do the remaining tasks at home. Answers will be published for all questions.

1. Illustrate the execution of the *heap–sort* algorithm on the following input sequence:

   $$(3, 15, 18, 5, 8, 6, 1).$$

   Consider both phases of the heap–sort: (i) insertion to the heap, and (ii) removing from the heap. Show how the sequence and heap look like after each step of each of the phase.

2. (a) Insert into an empty binary search tree, entries with keys $(30, 40, 24, 58, 48, 26, 11, 13)$ (in this order). Show the final 8–element binary search tree (do not show the intermediate trees).

   (b) Present step by step the execution of method: $TreeSearch(58, root)$ on tree T.

   (c) Present step by step the execution of operations:
   — removal of node with key 58 from tree T,
   — removal of node with key 40 from tree T.
   — ADDITIONAL – removal of node with key 30 from tree T.

   Method *TreeSearch(k,v)* was presented during the Lecture 9. It is presented below:

```
1 Algorithm TreeSearch(k,v)
2    if isExternal (v)
3         return v
4    if k < key(v)
5         return TreeSearch(k,left(v))
6    else if k = key(v)
7         return v
8    else { k > key(v) }
9         return TreeSearch(k,right(v))
```

3. Draw the 13–entry hash table that results from using a hash function:

   $$h(i) = (i + 5) \mod 13,$$

   to hash the keys:

   $$5, 6, 11, 4, 1, 15, 14, 2, 17, 28, 24.$$

   Assume that collisions are handled by:

   (a) Double hashing using the following secondary hashing function: $h'(i) = 5 - (i \mod 5)$.

   (b) ADDITIONAL TASK — Linear probing.

   (c) ADDITIONAL TASK — Separate chaining.

4. Illustrate the execution of the *merge–sort* algorithm on the following input sequence:

   $$(15, 8, 9, 10, 1, 7, 25, 2, 8).$$

   Present an execution of merge–sort as a binary tree.

5. Illustrate the execution of the *quick–sort* algorithm on the following input sequence:

$$(15, 8, 9, 10, 1, 7, 25, 2, 8).$$

Present an execution of quick–sort as a binary tree. Choose the pivot to be the last element in the sequence.

6. Sort the following sequence using *bucket–sort* method:

$$(3, 7, 11, 10, 6, 3, 6, 9, 12).$$

Present both phases of bucket–sort: (i) phase 1 – move entries from a sequence into appropriate bucket in an array, (ii) phase 2 – move entries from buckets to the final sequence.

ADDITIONAL EXERCISES:

7. Suppose that each row of an $n \times n$ array A consists of 1's and 0's such that, in any row of A, all the 1's come before any 0's in that row. Assuming A is already in memory, describe a method running $O(n\log n)$ time (not $o(n^2)$ time!) for counting the number of 1's in A.

8. An airport is developing a computer simulation of air–traffic control that handles events such as landing and takeoffs. Each event has a time–stamp that denotes the time when the event occurs. The simulation program needs to efficiently perform the following two fundamental operations:

   - Insert an event with a given time–stamp (that is, add a future event).
   - Extract the event with the smallest time–stamp (that is, determine the next event to process).

   Which data structure should be used for the above operations? Why?

9. Explain why during the lecture 7, the case where node r has a right child but not a left child was not considered in the description of the down–heap bubbling.

10. How many *different binary search trees* can store the keys $1, 2, 3, 4$? Justify your answer.