CSCI2100: Midterm Solutions

Problem 1 (10%). $5n + 3\sqrt{n} \le 8n$ for all $n \ge 1$.

Problem 2 (12%). Use the k-selection algorithm to find the k_1-th smallest integer a_1 of S. Then, run the algorithm again to find the k_2-th smallest integer a_2 of S. Finally, scan S to report all the integers $x \in S$ satisfying $a_1 \le x \le a_2$.

Problem 3 (13%). Create a hash table on S_1 in $O(n)$ time. Then, for each $x \in S_2$, probe the hash table to see whether $x \in S_1$; if so, output it.

Problem 4 (10%). First, it is clear that 10 is the root. As $10 < 30$, we must be entering the right subtree of 10. As $30 > 20$, we must be entering the left subtree of 30. As the next number is 60, we know that 60 lies in the left subtree of 30. This is not possible in a BST.

Problem 5 (10%). The statement is correct. Let u (resp., v) be the left (resp., right) child node of the root. Denote by k the 2nd smallest integer of S. If k is at neither u nor v, then it must be the key of a proper descendant of either u or v. In the former case, k is smaller than the key of u, which is not possible in a heap. Similarly, the latter case is not possible, either.

Problem 6 (15%). Initialize an empty linked list Q. At all times, Q stores its numbers in ascending order. To perform an `ins-large`(e), we check if e is larger than the integer at the tail of Q. If not, ignore e; otherwise, append e to the end of Q. To perform a `del-min`, simply remove the first element of Q.

Problem 7 (15%). We first solve the following "two-sum" problem (in fact, the problem was discussed in a tutorial). Given an integer t, determine if there exist distinct $i, j \in [1, n]$ such that $A[i] + A[j] = t$. For this purpose, first create a hash table on the elements in A, which takes $O(n)$ time. Then, for each $i \in [1, n]$, check whether $t - A[i]$ exists in the hash table, which takes $O(1)$ expected time. Doing so for all $i \in [1, n]$ requires $O(n)$ expected time.

To solve the original problem, we will deal with n instances of the two-sum problem. Specifically, for each $i \in [1, n]$, check whether there exist distinct $j, k \in [1, n] \setminus \{i\}$ such that $A[j] + A[k] = t - A[i]$. As discussed, each instance incurs $O(n)$ expected time. The total amount of time is thus $O(n^2)$ in expectation.

Problem 8 (15%). Initialize an empty min-heap H'. Insert the key of the root of H into H'. Repeat the following procedure k times.

- Perform a del-min from H'. Denote by u the node removed by the del-min operation. Report the key of u.
- Let v_1 and v_2 be the child nodes of u in H. Insert v_1 and v_2 to H'.

The heap H' has at most k elements (every del-min on H' is followed by at most two insertions into H'). Therefore, every execution of the above procedure takes $O(\log k)$ time, resulting in $O(k \log k)$ time overall.