Data Structures and Algorithms

Practice Sheet

Topic: *Binary Trees beyond searching and sorting*

1. You are given a sequence $\langle x_0, \ldots, x_{n-1} \rangle$ of n numbers. Design a suitable data structure which can perform each of the following operations in $O(\log n)$ time for any $0 \leq i \leq j < n$.

   - *Report_sum*(i, j): Report the sum of all numbers $\{x_i, ..., x_j\}$.
   - *Update*(i, Δ): Add Δ to the current value of x_i.

2. You are given a sequence $S = \langle b_0, \ldots, b_{n-1} \rangle$ of n bits. Design a suitable data structure which can perform each of the following operations in $O(\log n)$ time for any $0 \leq i < n$.

   - *Report_longest_sequence*: Report the length of the longest contiguous subsequence of 1s in the sequence S.
   - *Flip_bit*(i): Flip bit b_i.

3. You are given an initial sequence $\langle x_0, \ldots, x_{n-1} \rangle$ of n numbers. Design an efficient data structure for maintaining the sequence under the following operations. You need to guarantee $O(\log n)$ time complexity for each operation.

   - *Report(i)*: Report the i th element of the sequence.
   - *Insert*(i, x): Insert an element x at ith place in the sequence.
   - *Delete*(i): Delete ith element of the sequence.

   Hint: Use a red-black tree. What should be the key used for the red-black tree ? You will have to suitably augment each node suitably with additional information so as to help you perform the above operations efficienttly.