

Assignment 3

Foundations of Algorithms

1. [30 pts, divide and conquer]

You are given K number of coins with equal weights and a **mechanical balance scale** that can show if the pans (and their loads) are balanced or not. The problem is posed as finding the single different coin in the collection, which was mixed up accidentally. Devise the quickest algorithm, which can find the different coin. Generalize your solution for any K . Show the recurrence relation, time and space complexity.



2. [40 pts, RB trees]

Implement an algorithm to compute the height of each of the Red Black tree nodes. Then empirically show that an RB tree has an average height of $\log n$ which is also the complexity of the RB tree search function, i.e. $O(\log n)$. Your implementation of height computation should be efficient with $O(n)$ complexity. You can utilize the provided Python RB tree script in the lecture notes.

3. [30 pts, divide and conquer]

Solve exercise 9-3-8 (page 223) given two sorted lists X and Y , find the median of the elements in X and Y combined (Note that, it is not necessary to append or merge X and Y , but the algorithm will treat them separately.)

