

Problem Set 4 (due Monday, March 7)

1. (4 points) Insertion and deletion in binary search trees

Is the operation of insertion in binary search trees “commutative” in the sense that inserting x and then y into a binary search tree leaves the same tree as inserting y and then x . Argue why it is or give a counterexample. Similarly, determine whether the operation of deletion is commutative.

2. (8 points) Radix trees

Problem 12-2, pages 269–270.

3. (6 points) Hashing techniques

Exercise 11.4–1, page 244.

4. (5 + 5 = 10 points) Quadratic probing

Problem 11-3, pages 250-251.

5. (5 + 6 + 1 = 12 points) Coin changing

Problem 16-1, parts (a), (b), and (c), page 402.

6. (10 points) Human resource allocation

Suppose you are a manager in a construction firm and you are managing n building projects. You are asked to assign m of the engineers in your firm among these n projects. Assume for simplicity that all of the engineers are equally competent.

After some careful thought, you have figured out how much benefit i engineers will bring to project j . View this benefit as a number. Formally put, for each project j , you have computed an array $A_j[0..m]$ where $A_j[i]$ is the benefit obtained by assigning i engineers to project j . Assume that $A_j[i]$ is nondecreasing with increasing i . Further make the (plausibly sound) assumption that the marginal benefit obtained by assigning an i th engineer to a project is nonincreasing as i increases. Thus, for all j and $i \geq 1$, $A_j[i + 1] - A_j[i] \leq A_j[i] - A_j[i - 1]$.

Design a greedy algorithm to determine how many engineers you will assign to each project such that the total benefit obtained over all projects is maximized. Justify the correctness of your algorithm and analyze its running time.