

CS 457, Data Structures and Algorithms I

Second Problem Set

October 7, 2016

Due on October 14. Collaboration is not allowed. Contact me directly with any questions.

1. (20 pts) Solve problem 4-3 on page 108 of your textbook.
2. (15 pts) You are given a set S of n integers, as well as one more integer v . Design an algorithm that determines whether or not there exist two distinct elements $x, y \in S$ such that $x + y = v$. Your algorithm should run in time $O(n \log n)$, and it should return (x, y) if such elements exist and (NIL, NIL) otherwise. Prove the worst case running time bound and the correctness of the algorithm.
3. (15 pts) Prove tight **worst-case** asymptotic upper bounds for the following recurrence equations that satisfy $T(n) = 1$ for $n \leq 2$, and depend on a variable $q \in [0, n/4]$.
 - (a) $T(n) = T(n - 2q - 1) + T(3q/2) + T(q/2) + \Theta(1)$
 - (b) $T(n) = T(n - q - 1) + T(n/2 - q) + \Theta(n)$
 - (c) $T(n) = T(n - q - 1) + T(3q) + \Theta(n)$
4. (10 pts) Consider the following silly randomized variant of binary search. You are given a sorted array A of n integers and the integer v that you are searching for is chosen uniformly at random from A . Then, instead of comparing v with the value in the middle of the array, the randomized binary search variant chooses a random number r from 1 to n and it compares v with $A[r]$. Depending on whether v is larger or smaller, this process is repeated recursively on the left sub-array or the right sub-array, until the location of v is found. Prove a tight bound on the expected running time of this algorithm.
5. (10 pts) Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \log n$? Why or why not? Give an asymptotic upper bound for this recurrence.
6. (10 pts) Use a recursion tree to give an asymptotically tight solution to the recurrence:
 - (a) $T(n) = T(n - a) + T(a) + cn$, where $a \geq 1$ and $c > 0$ are constants.
 - (b) $T(n) = T(an) + T((1 - a)n) + cn$, where $a \in (0, 1)$ and $c > 0$ are constants.
7. (10 pts) For the following recurrences, find the bound implied by the master theorem. Then, try to prove the same bound using the substitution method. If your initial approach fails, show how it fails, and try subtracting off a lower-order term to make it work:
 - (a) $T(n) = 4T(n/3) + n$
 - (b) $T(n) = 4T(n/2) + n$
8. (5 pts) Solve the recurrence $T(n) = 3T(\sqrt{n}) + \log n$ by making a change of variable. Your solution should be asymptotically tight. Do not worry about whether values are integral.
9. (5 pts) Using proof by induction, show that there are at most $\lceil n/2^{h+1} \rceil$ nodes of height h in any n -element heap.