

Fall 2018, CMPSC 465: Exam 2.

Closed book and closed notes, no ‘cheat sheet’, no calculators allowed.

Please don’t use cell phones during the exam.

Answer questions in the space provided.

The exam is for 40 points (4 problems, 10 points each).

NAME:

1. A binary tree is full if all of its vertices have either zero or two children. Let B_n denote the number of full binary trees with n vertices. For general n , give a recurrence relation for B_n . Briefly explain how you arrived at this recurrence. Using the recurrence, give the values for B_8 and B_9 .

Solution:

2. An array $A[1 \dots n]$ is said to have a *majority element* if more than half of its entries are the same. Given an array, design an $O(n \log n)$ divide-and-conquer algorithm to determine if the array has a majority element, and, if so, to find that element.

Solution:

3. Consider the task of searching a sorted array $A[1 \dots n]$ for a given element x : a task we usually perform by binary search in time $O(\log n)$. Show that any algorithm that accesses the array only via comparisons must take $\Omega(\log n)$ steps.

Solution:

4. The square of a matrix A is its product with itself, AA .
- (a) Show that five multiplications are sufficient to compute the square of a 2×2 matrix.
 - (b) Is the following a valid algorithm for computing the square of an $n \times n$ matrix? If yes, briefly illustrate its working. If not, explain why.
“Use a divide-and-conquer approach as in Strassen’s algorithm, except that instead of getting 7 subproblems of size $n = 2$, we now get 5 subproblems of size $n = 2$ thanks to part (a). Using the same analysis as in Strassen’s algorithm, we can conclude that the algorithm runs in time $O(n^{\log_2 5})$.”

Solution: