

CS251 Homework 1

Handed out: Feb 20, 2017

Due date: Feb 27, 2017 at 11:59pm (This is a **FIRM** deadline, solutions will be released immediately after the deadline)

Question	Topic	Point Value	Score
1	True / False	5	
2	Match the Columns	7	
3	Short Answers	22	
4	Programming Questions	22	
5	Symbol Tables	4	
Total		60	

1. True/False [5 points]

1. Amortized analysis is used to determine the worst case running time of an algorithm.
2. An algorithm using $5n^3 + 12n \log n$ operations is a $\Theta(n \log n)$ algorithm.
3. An array is partially sorted if the number of inversions is linearithmic.
4. Shellsort is an unstable sorting algorithm.
5. Some inputs cause Quicksort to use a quadratic number of compares.

2. Match the columns [7 points]

- | | | |
|--------------------|-----------------------|---|
| A. Mergesort | <input type="radio"/> | <input type="radio"/> 1. Works well with duplicates |
| B. Quicksort | <input type="radio"/> | <input type="radio"/> 2. Optimal time and space |
| C. Shellsort | <input type="radio"/> | <input type="radio"/> 3. Works well with order |
| D. Insertion sort | <input type="radio"/> | <input type="radio"/> 4. Not analyzed |
| E. Selection sort | <input type="radio"/> | <input type="radio"/> 5. Stable and fast |
| F. 3-way quicksort | <input type="radio"/> | <input type="radio"/> 6. Optimal data movement |
| G. Heapsort | <input type="radio"/> | <input type="radio"/> 7. Fast general-purpose sort |

3. Short Answers [22 points]

- (a) Suppose that the running time $T(n)$ of an algorithm on an input of size n satisfies $T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + cn$ for all $n > 2$, where c is a positive constant. Prove that $T(n) \sim cn \log_2 n$. [4 points]

(b) Rank the following functions in increasing order of their asymptotic complexity class. If some are in the same class indicate so. **[4 points]**

- $n \log n$
- $n^2/201$
- n
- $\log^7 n$
- $2^{n/2}$
- $n(n-1) + 3n$

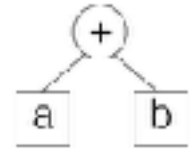
(c) Consider the following code fragment for an array of integers:

```
int count = 0;
int N = a.length;
Arrays.sort(a);
for (int i = 0; i < N; i++)
    for (int j = i+1; j < N; j++)
        for (int k = j+1; k < N; k++)
            if (a[i] + a[j] + a[k] == 0)
                count++;
```

Give a formula in tilde notation that expresses its running time as a function of N. If you observe that it takes 500 seconds to run the code for N=200, predict what the running time will be for N=10000. **[5 points]**

(d) In Project 2 you were asked to use `Arrays.sort(Object o)` because this sort is stable. What sorting algorithm seen in class is used in this case? What sorting algorithm would you use if instead of dealing with `Point` objects you were handling `float` values? Justify your answer. **[4 points]**

- (e) Convert the following (*Infix*) expressions to *Postfix* and *Prefix* expressions
(To answer this question you may find helpful to think of an expression "a + b"
as the tree below.) **[5 points]**



(i) $(a + b) * (c / d)$

(ii) $a * (b / c) - d * e$

(iii) $a + (b * c) / d - e$

(iv) $a * b + c * (d / e)$

(v) $a * (b / c) + d / e$

4. Programming Questions [22 points]

- (a) Give the pseudocode to convert a fully parenthesized expression (*i.e.*, an INFIX expression) to a POSTFIX expression and then evaluate the POSTFIX expression.
[5 points]

- (b) Given two sets A and B represented as sorted sequences, give Java code or pseudocode of an efficient algorithm for computing $A \oplus B$, which is the set of elements that are in A or B, but not in both. Explain why your method is correct.
[5 points]

(c) Let A be an unsorted array of integers $a_0, a_1, a_2, \dots, a_{n-1}$. An inversion in A is a pair of indices (i, j) with $i < j$ and $a_i > a_j$. Modify the merge sort algorithm so as to count the total number of inversions in A in time $\mathcal{O}(n \log n)$. [**5 points**]

(d) Let $A[1 \dots n]$, $B[1 \dots n]$ be two arrays, each containing n numbers in sorted order. Devise an $\mathcal{O}(\log n)$ algorithm that computes the k -th largest number of the $2n$ numbers in the union of the two arrays. Do not just give pseudocode — explain your algorithm and analyze its running time.

For full credit propose a solution using constant space. [**7 points**]

5. Symbol Tables [4 points]

Draw the Red-Black LL BST obtained by inserting following keys in the given order:
H O M E W O R K S.