

CMPS 101

Homework Assignment 1

1. p.29: 2.2-2
Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A . Write pseudo-code for this algorithm, which is known as *selection sort*. What loop invariant does this algorithm maintain? Why does it need to run for only the first $n-1$ elements, rather than for all n elements? Give the best-case and worst-case running times of selection sort in Θ -notation.
2. p.37: 2.3-1
Using Figure 2.4 as a model, illustrate the operation of merge sort on the array $A = (3, 41, 52, 26, 38, 57, 9, 49)$.
3. p.39: 2.3-4
We can express insertion sort as a recursive procedure as follows. In order to sort $A[1 \dots n]$, we recursively sort $A[1 \dots n-1]$ and then insert $A[n]$ into the sorted array $A[1 \dots n-1]$. Write a recurrence for the running time of this recursive version of insertion sort.
4. p.39: 2.3-5
Referring back to the searching problem (see Exercise 2.1-3), observe that if the sequence A is sorted, we can check the midpoint of the sequence against v and eliminate half of the sequence from further consideration. *Binary search* is an algorithm that repeats this procedure, halving the size of the remaining portion of the sequence each time. Write pseudo-code, either iterative or recursive, for binary search. Argue that the worst-case running time of binary search is $\Theta(\lg n)$.
5. p.41-42: 2-4abcd
Let $A[1 \dots n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A .
 - a. List the five inversions of the array $(2, 3, 8, 6, 1)$.
 - b. What array with elements from the set $\{1, 2, 3, \dots, n\}$ has the most inversions? How many inversions does it have?
 - c. What is the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer.
 - d. Give an algorithm that determines the number of inversions in any permutation of n elements in $\Theta(n \lg n)$ worst-case time. (Hint: Modify merge sort.)