Weekplan: Searching and Sorting

Philip Bille Inge Li Gørtz

Reading

Introduction to Algorithms, 4th edition, Cormen, Rivest, Leisersons, and Stein (CLRS): Chapter 2.

Exercises

1 Run by Hand and Properties Solve the following exercises.

1.1 [w] Show the execution of insertion sort on the array $A = [31, 41, 59, 26, 41, 58]$.

1.2 [w] Modify the pseudocode for insertion sort to sort the input array in non-decreasing order instead of non-increasing order.

1.3 [w] Show the execution of merge sort on the array $A = [3, 41, 52, 26, 38, 57, 9, 49]$

1.4 Convince yourself that insertion sort may be expressed recursively as follows: to sort $A[0, n-1]$ we recursively sort $A[0, n-2]$ and then insert $A[n-1]$ into to the sorted array $A[0, n-2]$. Write a recurrence for the running time and then find a solution.

1.5 A friend suggest that you should use binary search to speed up the insertion step in insertion sort. Will this work and if so, how will it affect the running time of the algorithm?

2 Duplicates and Close Neighbours Let $A[0..n-1]$ be an array of integers. Solve the following exercises.

2.1 [w] A *duplicate* in A is a pair of entries i and j such that $A[i] = A[j]$. Give an algorithm that determines if there is a duplicate in A in $O(n^2)$ time.

2.2 Give an algorithm that determines if there is a duplicate in A in $O(n \log n)$ time. *Hint:* use merge sort.

2.3 A *closest pair* in A is a pair of entries i and j such that $|A[i] - A[j]|$ is minimal among all the pairs of entries. Give an algorithm that finds a closest pair in A in $O(n \log n)$ time.

3 [†] **Stones** Josefine likes to go to the beach and collect stones. Josefine likes to bring home as many stones as possible, but she can only carry W kilograms of stones. Give an algorithm that, given a list of the weights of N stones and the maximum weight W, determines the maximal number of stones she can bring home that day.

4 Correctness of Merge Sort Show that merge sort sorts all arrays correctly. You can assume that merge correctly merges sorted arrays. *Hint:* use induction.

5 2Sum and 3Sum Let $A[0..n-1]$ be an array of integers (positive and negative). The array A has a 2-*sum* if there exist two entries i and j such that $A[i] + A[j] = 0$. Similarly, A has a 3-*sum* if there exists three entries i, j and k such that $A[i] + A[j] + A[k] = 0$. Solve the following exercises.

5.1 [w] Give a simple algorithm that determines if A has a 2-sum in $O(n^2)$ time.

5.2 Give an algorithm that determines if A has a 2-sum in $O(n \log n)$ time. *Hint:* use binary search.

5.3 [w] Give an algorithm that determines if A has a 3-sum in $O(n^3)$ time.

5.4 Give an algorithm that determines if A has a 3-sum in $O(n^2 \log n)$ time. *Hint:* use binary search.

5.5 [$**$] Give an algorithm that determines if A has a 3-sum in $O(n^2)$ time.

6 Selection, Partition, and Quick Sort Let $A[0..n-1]$ be an array of distinct integers. The integer with *rank* k in A is the kth smallest integer among the integers in A. *The median* of A is the integer in A with rank $\lfloor (n-1)/2 \rfloor$. Solve the following exercises.

6.1 Give an algorithm that given a k finds the integer with rank k in A in $O(n \log n)$ time.

A *partition* of A is a separation of A into two arrays A_{low} and A_{high} such that A_{low} contains all integers from A that are smaller than or equal to the median of A and A_{high} contains all the integers from A that are larger than the median of A. Assume in the following that you are given a linear time algorithm to determine the median of an array.

6.2 Give an algorithm to compute a partition of A in $O(n)$ time.

6.3 $[*]$ Give an algorithm to sort A in $O(n \log n)$ time using recursive partition.

6.4 $[**]$ Give an algorithm that given a k finds the integer with rank k in A in $O(n)$ time.