S. Schwoon / D. Suwimonteerabuth Winter Term 2008/09

Fundamental Algorithms

Exercise Sheet 4

1. Let a be an array of integers of length n and s be an integer between 1 and n. We define the s-th smallest element of a to be the element in the s-th position if a is sorted in nondecreasing order. Given a and n, the *selection* problem is the problem of finding the s-th smallest element of a. The *median* of a is defined as its $\lceil n/2 \rceil$-th smallest element. For instance, if a is $[6, 2, 7, 4, 8, 9, 1, 1, 2]$, then its median is 4.

   Obviously, any algorithm that solves the selection problem can also be used to find the median: simply select the $\lceil n/2 \rceil$-th smallest. We consider the opposite in this exercise, i.e. how can one solve the selection problem of an array, given an algorithm that finds its median?

   We recall from the lecture the procedure `partition` used by the algorithm `quicksort`. Given an array a and indices `lo` and `hi`, the procedure `partition` picks a pivot p from a, and partitions a into three parts: (i) $a[\texttt{lo}, \texttt{left} - 1]$ having values less than p, (ii) $a[\texttt{left}, \texttt{eq} - 1]$ having values equal to p, and (iii) $a[\texttt{eq}, \texttt{hi}]$ having values greater than p. Therefore, the element we want to find must be in one of the three subarrays. If it is in the second array, we are done. Otherwise, we can delimit the search range to a subarray, and continue the search from there.

   (a) Ignoring the problem of finding median for now, give an algorithm `selection` that solves the selection problem by calling the procedure `partition` repeatedly. Assume that pivots are always the first elements of the array. Analyze the complexity of your algorithm, both on averge and in the worst case.

   (b) Assume that there is an algorithm `median5` that finds the median of an array of length at most 5. Devise the algorithm `medianofmedians` that divides an array into $\lfloor n/5 \rfloor$ blocks of length 5, computes medians for each block, and computes again the mediean of medians.

   (c) Modify the algorithm `selection` in (a) such that `medianofmedians` is used for finding pivots. Analyze the complexity of `selection` in this case.

2. Let a and b be integers having n digits. *Long multiplication* is a natural way of multiplying a by b taught in schools: multiply a by each digit of b and then add up all the properly shifted results. The operations needed are multiplications for single digits, shifts, and additions. The time complexity is obviously $\Theta(n^2)$.

   Devise a multiplication algorithm based on the divide-and-conquer paradigm that has a better complexity, using only three operations above.

   Hints: if n is an even number, we can rewrite $a = 10^{n/2}u + v$ where u and v are first and last $n/2$ digits of a, respectively, and $b = 10^{n/2}x + y$, where x and y are first and last $n/2$ digits of b, respectively.