Datenstrukturen & Algorithmen          Exercise Sheet 3           FS 12

Exercise 3.1  Sorting methods.

Answer the following questions and give a brief explanation of your response:

a) Is the sorted sequence $1, 2, \ldots, n$ a Min-Heap?

b) When all the elements in a Max-Heap are different, at which positions could the smallest element be found?

c) In an array with no equal elements, with which probability does Quicksort with random Pivot select the smallest or the largest element in each step, such that it always makes the worst split? How large is this probability as a percentage for $n = 10$? And for $n = 20$?

d) A comparison-based algorithm is called stable if the relative order of identical elements is not changed. If the element '5', for example, occurs twice in an array, then the first 5 is never moved past the second 5. Which of the comparison-based sorting methods that you know are stable, or can easily be adapted accordingly?

e) A sorting algorithm is called in-situ if it works on the input sequence using only a constant amount of additional space for storing parts of the sequence. Which sorting algorithms that you know are in-situ, or can easily be adapted accordingly?

f) The worst-case running time of the Quicksort is $\Theta(n^2)$, while the worst-case running time of the Mergesort is $\Theta(n \log n)$. Guess two (good) reasons why, despite this fact, the Quicksort is the most used solution in practice (e.g. sort in the C++ standard library).

g) In some textbooks, Radixsort is called a ,,linear-time sorting algorithm“. Does this make sense when sorting sequences without any two identical numbers? Identify a condition as weak as possible such that any sequence of numbers that satisfies this condition is sorted in $O(n)$ steps by Radixsort.

Exercise 3.2  Various topics (Part of an exam in August 2011).

a) The integer multiplication method of Karatsuba/Ofman computes the product of two numbers recursively using a formula that, except for additions and multiplications with the base (here: 10), contains three products. Give two numbers $x$ and $y$ such that these products are, respectively, $(74 \cdot 93)$, $(51 \cdot 80)$ and $(74 \pm 80) \cdot (51 \pm 93)$.

\[
x = \underline{\hspace{2cm}} \quad y = \underline{\hspace{2cm}}
\]

b) Give a sequence of 5 numbers such that Bubblesort needs exactly 10 swaps to sort it.

Answer: \underline{\hspace{7cm}}

c) The following array contains the element of a Min-Heap stored in the usual form. How does the array appear after the minimum is removed, and the heap condition is restored?

\[
\begin{array}{cccccccc}
25 & 60 & 32 & 61 & 62 & 52 & 57 & 80 & 86 \\
\end{array}
\]
Exercise 3.3  Algorithm design: sums of numbers.

In this exercise we will design efficient algorithms for the following questions. An array $A[1..n]$ of natural numbers is given as input.

a) Given a natural number $z$, find two numbers $a$ and $b$ in $A$ such that $a + b = z$, if such elements exist. Suggest the most efficient algorithm for this problem. Which is its running time?

b) Suppose now that the array $A$ is sorted in ascending order. How efficiently can the problem in a) be solved in this case? Give an algorithm with that worst-case running time. 

Hint: It is possible to achieve a better running time than the one in part a).

c) A more difficult variation is the following: Does $A$ contain three different numbers $a, b, c$ such that $a + b = c$? Design and describe the most efficient algorithm for this problem, and specify its running time.

Hand-in: until Wednesday 14th March 2012.