TECHNISCHE UNIVERSITÄT BERLIN
Institut für Mathematik

Dr. Rüdiger Stephan

Mathematical Tools for Engineering and Management
Winter Term 2011/2012

October 19, 2011

Exercise session 1

Exercise 1

In this exercise we consider the problem of sorting numbers and discuss three sorting algorithms, *Insertion Sort*, *Selection Sort*, and *Merge Sort*.

1. We write up Insertion Sort in PSEUDOCODE.

   InsertionSort($(a_1, a_2, \ldots, a_n)$)
   Input: A sequence of numbers $(a_1, a_2, \ldots, a_n)$
   Output: A sequence of the same numbers in nondecreasing order

   ```
   FOR i := 2 TO n DO
       j := i;
       WHILE j ≥ 2 AND a_{j-1} > a_j DO
           // Swap, if necessary.
           b := a_j;
           a_j := a_{j-1};
           a_{j-1} := b;
           j := j - 1;
       ENDWHILE
   ENDFOR
   ```

   - Execute the algorithm for the instance $(2, 1, 5, 1, 3, 5, 7, 9)$.
   - Try to analyze the perfomance of the algorithm. What is the **worst case**? How many steps does the algorithm do? What would you say about the average performance?

2. Write up Selection Sort in PSEUDOCODE, execute the algorithm for the instance $(2, 1, 5, 1, 3, 5, 7, 9)$, and analyze the perfomance of the algorithm.

3. Execute Merge Sort for the instance $(2, 1, 5, 1, 3, 5, 7, 9)$ and analyze its perfomance.

4. Discuss advantages and disadvantages of the three algorithms.

Exercise 2

Look again at the Paint Shop model from the lecture.

1. What's an instance? A solution? The input?

2. Think about preprocessing: When you can decrease the input size without changing the (configuration of your) problem?

3. Come up with a simple rule to choose the next storage line. Analyse your rule: Is it efficient? What is the worst case that could happen?

4. Implement an (approximation) algorithm in PSEUDOCODE using your retrieval rule.

5. Compute lower bounds for the minimal number of color changes.