Datastructures and Algorithms: Exam 1

October 25, 2016

You are allowed to use copies of the slides during the exam.

1. Provide a pseudocode description for the operation $insertBefore(v, d)$ on singly-linked lists. (20 pts)

2. Apply heapsort to sort the following sequence of numbers.
   
   6 17 2 21 9 11 20 4
   
   Show how the heap datastructure evolves during the computation. (18 pts)

3. Explain why the time complexity of radix sort applied to a list of $n$ numbers with at most $k$ digits is $\Theta(k \cdot n)$. Discuss briefly how this compares to the lower bound of $O(n \cdot \log n)$ for the worst-case time complexity of sorting algorithms based on pairwise comparisons. (12 pts)

4. Apply Kosaraju’s algorithm to find the strongly connected components in the following directed graph.
Consider two different scenario’s, where the first depth-first search starts in node \( r \) or in node \( u \). (16 pts)

5. Suppose that Dijkstra’s shortest path algorithm is adapted as follows. For each neighbor \( w \in H \) of the root node \( v \), check whether

\[
d(v) + \text{weight}(wv) \leq d(w).
\]

Note that the \(<\) sign has been replaced by a \(\leq\) sign.

(a) Show on a small weighted graph that this may produce a different result, in comparison to the original Dijkstra’s shortest path algorithm. (9 pts)

(b) Is the resulting algorithm still correct? If yes, explain why. If not, give an example where the adapted algorithm produces a wrong result. (9 pts)

6. Show how the greedy approach computes a solution for the following instance of the 0-1 knapsack problem. There are four items \((10, 3)\) and \((8, 2)\) and \((11, 4)\) and \((6, 2)\), where the first index is the value and the second index is the weight of the item; the knapsack can carry no more than weight 8.

Is the solution you obtain optimal? (16 pts)