TDDB56 DALGOPT – Algoritmer och Optimering  
Kontrollskrivning DALG 2006-10-19, 8–13

Examinator: Jan Maluszynski  
Jour: Jan Maluszynski, 013-281483 eller 0707-524408  
Maxpoäng: 20p (som kan tillgodoräknas på de tre närmaste följande ordinarie tentamina i kursen om så önskas - detaljerade anvisningar om detta se kursensidan.)  
Tillåtna hjälpmedel: – Engelsk-svensk ordbok  
– Goodrich, Tamassia: *Data Structures and Algorithms in Java*  
– Cormen, Leiserson, Rivest: *Introduction to Algorithms*

Generella instruktioner:

- Läs igenom alla uppgifter innan du börjar.
- Redovisa maximalt en uppgift per inlämnat ark. Består uppgiften av flera deluppgifter kan dessa redovisas på samma ark. Skriv nämn och personnummer överst på varje ark.
- Skriv tydligt. Oläsbara lösningsförslag beaktas icke.
- Motivera tydligt alla steg i svaret/lösningen. Avskunad av motivering kan medföra poängavdrag.
- The solutions can be written in English or in Swedish, as you prefer.

Lycka till!
1. Complexity
   (a) Which of the following are true and which are false? Answers without justification give no points.
      - $\sqrt{n^3} \in O(n^2)$
      - $\min(700, n^2) \in \Theta(1)$
      - $n^3 \log n \in \Omega(n^3)$
      - $3^{n-2} \in O(2^n)$
   (b) Prove that if $f \in O(g)$ and $g \in O(h)$ then $f \in O(h)$. Hint: Show how the constants $n_0$ and $c$ in the definition of $f \in O(h)$ can be computed from the assumption that $f \in O(g)$ and $g \in O(h)$.

2. Queues and Stacks.
   Describe how to implement the ADT queue using two stacks. For obtaining maximal number of points the proposed solution should be as efficient as possible.
   (a) Explain the idea. Describe it in pseudocode using as basic operations the operations of ADT stack.
   (b) Assume that you enqueue and dequeue $n$ data items. Thus altogether we perform $n$ enqueue operations and $n$ dequeue operations. The order of these operations is such that the queue becomes empty only after the last dequeue operation. What is the worst case time for a single enqueue operation and a single dequeue operation? What is the amortized time of all these operations? Answers without justification give no points.

3. Implementations of ADT Map
   On an initially empty Map $M$ we execute the following sequence of operations: put(15, c), put(8, a), put(14, b), remove(15), put(32, d), put(4, e), put(7, f). Show how these consecutive operations are executed on $M$ implemented as:
   (a) a 7-element hash table (indexed 0 to 6) with open addressing and double hashing, where $h_1(k) = k \mod 7$, $h_2(k) = 5 - (k \mod 5)$, and the removal operation is based on marking the deleted entries.
   (b) a skip list where the random sequence used for controlling all the put operations is 1001110101100, where 1 indicates increase of the insertion level.

4. Search Trees
   (a) Consider the binary search tree whose preorder traversal gives the following sequence of keys: 41 16 39 67 49 47 61 74.
      - Draw the tree and prove that it is an AVL tree.
      - Show step by step how the AVL-tree operations Insert(40), Insert(53), Delete(67) are performed on this tree (each of the operations is to be applied to the original tree).
   (b) Show the (2,3) tree obtained from the empty tree by consecutive insertions of the keys 41, 16, 39, 67, 49, 47.
5. Sorting. 
Illustrate consecutive steps of sorting of the integer array [7, 5, 9, 6, 8] using the in-place variants of the following algorithms (as discussed in the course book). To get full points, the steps are to be clarified by short explanations.

(a) Insertion Sort. Is this algorithm stable? Justify your answer. 
(b) Quick Sort where the pivot is always chosen as the last element in the sequence. 
(c) Heap Sort using max-heap. Illustrate the steps both by array- and tree-representations.

6. Graphs
Peter is building a holiday house. He has broken the project into the following tasks: (1) preparing the ground, (2) building exterior walls, (3) building interior walls, (4) building the chimney, (5) building the roof, (6) painting the interior walls, (7) laying the floor, (8) setting doors, (9) setting windows, (10) covering the roof, (11) insulating the exterior walls, (12) moving in.

The restrictions on the ordering of these tasks are:

- The ground must be prepared before building the exterior walls or the chimney.
- To build the roof Peter must first build the exterior walls and the chimney.
- The interior walls are built after the exterior walls.
- The windows are set after building the exterior walls.
- The doors are set after building the interior walls.
- The floor can only be laid when the roof is covered and the interior walls are built.
- To cover the roof Peter must first build it.
- The insulation of exterior walls can only be done when the windows and the doors are already set.
- Painting of the interior walls can only be done when the floor is ready.
- Peter will not move in until the interior walls are painted and the exterior walls are insulated.

(a) Represent the above restrictions as a directed graph with vertices representing tasks and edges representing their ordering. The graph must not introduce more direct restrictions on task ordering than those stated above. 
(b) Show the sequence of nodes obtained during the depth-first search of the graph starting in node (1) that selects the adjacent nodes in the increasing order of the numbers assigned to the tasks. 
(c) Show the sequence of nodes obtained during a breadth-first search of the graph starting in node (1) and selecting the adjacent nodes in the same way as in 6b. 
(d) Peter wants to perform each of the tasks during consecutive weekends. Which of the graph algorithms discussed in the course can be used to schedule the tasks in the way that all restrictions are observed. Explain in detail how this algorithm works for your graph by discussing the intermediate stages of its operation and show the resulting schedule. Is the schedule unique?