# Algorithms and Data Structures – Midterm

1. We know that $T(n) \leq T(n-1) + n/3$, if $n \geq 105$ and $T(n) \leq 100$, if $n < 105$.
   Is it true that
   (a) (5 points) $T(n) = O(n^2)$ ?
   (b) (2 points) $T(n) = O(n^3)$ ?
   (c) (3 points) $T(n) = \Omega(n)$ ?

2. The array representation of a heap is: 2, 31, 6, 50, 51, 7.
   (a) (2 points) What is the tree-like version of this heap?
   (b) (4 points) Insert 4 into this heap.
   (c) (4 points) Perform extract_min in the original heap.

3. (10 points) Use Kruskal's algorithm with UNION-FIND data structure with path-compression from vertex $g$ to find a MST of the following undirected graph:
   **a**:b(2),c(3); **b**:a(2),d(2); **c**:a(3),d(1); **d**:b(2),c(1),e(2),f(4); **e**:d(2),f(1),g(2); **f**:d(4),e(1),g(2),h(1); **g**:e(2),f(2),h(3); **h**:f(1),g(3).

   In each step give the necessary operations we have to perform (FIND or UNION, for which input) and indicate how the data structure changes meanwhile.

4. (10 points) Reconstruct the binary tree whose inorder traversal gives $j, b, k, g, i, a, c, d, f, e, h$, and whose preorder traversal gives $a, b, j, g, k, i, d, c, e, f, h$.

5. (10 points) An unweighted directed graph $G = (V, E)$, and two vertices $s, t \in V$ are given. Vertices of $G$ are labeled red, green, or white, both $s$ and $t$ are red. Show how to find a minimum length admissible path from $s$ to $t$ (or determine that none exists), when a (not necessarily simple) path is admissible if it contains at most one edge from a white vertex to a green one, all the other vertices on the path are red. The algo should finish in $O(n + m)$.

6. (10 points) A connected, undirected, edge-weighted graph $G$ is given by its adjacency list, each edge-weight is positive. Each vertex of $G$ represents either a shop or a warehouse, the edges represent roads, and the edge-weights represent the cost of reconstruction of the corresponding road. We would like to reconstruct some of the roads in such a way that for any shop $s$ there exist a warehouse $w$ such that there is a new path between $s$ and $w$. (I. e. any shop is reachable from at least one warehouse by using only new roads.) We would like to find the cheapest reconstruction possible.

   Design an algorithm for this task, the running time should be $O(m \log n)$.

# Optional problems

1. (5 points) Given an $n \times n$ matrix with integer elements, design an algorithm to decide whether the matrix has two identical rows. The algorithm should use $O(n^2 \log n)$ comparisons where by one comparison two integers can be compared.

2. (5 points) In an array $A$ containing $n$ different numbers two indices $i$ and $j$ form an inversion if $i < j$ but $A[i] > A[j]$. (For example the number of inversions in $2, 5, 1, 10, 3$ is 4.) Our task is to compute the number of inversions in a given array, the running time of the algorithm should be $O(n \log n)$ for an array of length $n$.