Examination in
TDT4120 Algorithms and Datastructures
Tuesday 3rd of August 2004, hrs. 0900-1500

Contact under examination: Magnus Lie Hetland tlf. 91851949
Legal means: All types of calculators. All written and printed material. Write your answers in the given places. An answer without explanation does not count. Use extra sheets only if necessary. Write "Student nr." on each sheet.

The problems are marked #n/p, where n is the problem id and p is the maximum obtainable points when both the (Yes/No) answer and the explanation is correct.

Maximum score sum for this examination is 50.

#1/1: Best case runtime for INSERTION SORT by sorting n elements is O(n).

Answer: Explanation:

#2/3By using the Master-theorem you find the solution T(n) = Θ (nlogn) for the recurrence T(n) = 3 T(n/3) + logn

Answer: Explanation:

#3/3: For a random binary search tree with n nodes we can print the nodes in sorted order in O(n) time.

Answer: Explanation:

#4/1: Any binary search tree with n nodes has height O(logn).

Answer: Explanation:

#5/1: Any heap used by HEAPSORT to sort n elements has height O(log n)

Answer: Explanation:

Student number:

#6/2: The heap in HEAPSORT is randomly ordered.

Answer: Explanation:

#7/2: It is such that $n\log n^2 = O(n^2)$.

Answer: Explanation:

#8/3: MERGESORT uses in worst-case $O(n^2)$ time.

Answer: Explanation:

#9/2: A breadth first search algorithm makes use of a stack.

Answer: Explanation:

#10/2: A depth-first search-algorithm makes use of a stack.

Answer: Explanation:

#11/3: A maximal-matching in a bipartit graph can be found by Linear Programming.

Answer: Explanation:

#12/3: If some edge weights in a graph G = (V,E) are negative, the shortest path from node s to node t can be found by using Dijkstra’s algoritme if we first add a big constant C to all of E’s lengths such that all these become non-negative.

Answer: Explanation:

#13/1: Any DAG (directed acyclic graph) can in a unique way be sorted topologically.

Answer: Explanation:

Student number:

#14/1: Dijkstras algorithm is an example of a greedy algorithm.

Answer: Explanation:

#15/1: : Dijkstras algorithm is an example of dynamic programmering.

Answer: Explanation:

#16/3: If all edge capacities of a flow-graph are multiples of of 5, then also the maximum flow will be a multiple of 5.

Answer: Explanation:

#17/4: In a flow-graph with nodes s,a,b,c,d,t we let f/m mean that f units flow on an edge with capacity m. We have the following edge-flows: (s,a):2/4, (s,c):6/6, (c,a):3/7, (a,b):5/5, (b,c):0/2, (c,d):3/3, (b,d):1/4, (d,t):4/4, (b,t):4/6. This edge-flow is maximal.

Answer: Explanation:

#18/3: Let P be the shortest path from s to t in a graph G=(V,E). If we increase the length of all edges in E by 1, P will still be the shortest path from s to t.

Answer: Explanation:

#19/3: A depth-first search in a graph is asymptotically quicker than a breadth-first search.

Answer: Explanation:

#20/2: n integers in the range $[0,n^{100}]$ can be sorted in linear time.

Answer: Explanation:

Student number:

#21/2: Graph G is acyclic if no ”back-edges” occur during a depth-first traversal of G.

Answer: Explanation:

#22/4: We shall merge k sorted lists, each with n/k elements, by the following method: Merge the 2 first lists; the result is merged with the third list, this result is further merged with the fourth list, and so on, until the last list of n/k elements is merged in. The claim is here that this algorithm requires Θ (kn) tid.

Answer: Explanation: