Some important points:

(*i*) Read the entire exam thoroughly *before* you start!

(*ii*) The teacher will normally take one round through the exam hall. Have your questions ready!

(*iii*) Write your answers in the answer boxes and hand in the problem sheet. Feel free to use a pencil! Or draft your answers on a separate piece of paper, to avoid strikouts, and to get a copy for yourself

(*iv*) You are permitted to hand in extra sheets if need be, but the intention is that the answers should fit in the boxes on the problem sheets. Long answers do not count positively.

The exam has 20 problem, worth 120 points in total.
Of these, 20 are bonus points, so a score above 100 counts as 100.
The point value is given next to each problem.

* * *

For each of the first 5 problems, you are to select your answer from among these algorithms, and and check the circle to the left of the correct letter. For each problem the answer may consist of multple algorithms, so you are permitted to check off multiple circles.

A Bucket-Sort

B Counting-Sort

C Heapsort

D Insertion-Sort

E Merge-Sort

F Quicksort

G Radix-Sort

H Randomized-Quicksort

I Randomized-Select

J Select

(5 p) 1. Which algorithms on p. 1 are based on the design method *divide-and-conquer*?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 2. Which algorithms on p. 1 have a running time of $\mathrm{O}(n)$ in the *worst case*, if $k = \mathrm{O}(n)$ and $d = \mathrm{O}(1)$?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 3. Which algorithms on p. 1 use the subroutine PARTITION, possibly slightly modified?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 4. Which algorithms on p. 1 use a priority queue?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 5. Which algorithms on p. 1 change running time from the best to the worst case?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 6. What is the solution to the recurrence $T(n) = 3T(n/3) + n$? Give your answer in O notation.

(5 p) 7. What is the solution to the recurrence $T(n) = T(n/5) + n$? Give your answer in O notation.

(5 p) 8. What is the solution to the recurrence $T(n) = 2T(4\sqrt{n}) + \lg n$? Give your anwer in O notation.

(5 p) 9. If you insert 1, 2, 9, 5, 10, 7, 6, 4, 8 and 3 into an empty binary tree (one by one, in this order), what is the height of the resulting tree (number of edges in longest path from root to a leaf)?

(5 p) 10. If you insert 1, 2, 9, 5, 10, 7, 6, 4, 8 and 3 into an empty binary min-heap (one by one, in this order), what is the height of the resulting heap (number of edges in longest path from root to a leaf)?

(5 p) 11. In a sorted table with n elements, an element x occurs once, while all other elements occur twice. Describe an efficient algorithm to determine x. What is the running time?

(5 p) 12. Very briefly explain what it means for a sorting algorithm to be *stable*.

(5 p) 13. You run DFS on a directed graph, so all edges are classified. How may you use this classification to determine whether the graph has a cycle?

(5 p) 14. You have a binary matrix indicating whether there are roads directly between various pairs of cities. You wish to construct a new matrix indicating whether there are direct *or indirect* roads between each pair, that i, if you are permitted to drive via other cities. Assume that there are n cities. How would you solve the problem if there are $\Theta(n)$ direct roads? What is the running time?

(5 p) 15. How would you solve the problem in 14 if there are $\Theta(n^2)$ direct roads? With what running time?

(5 p) 16. Draw the residual network of the flow network drawn in Fig. 1 (see p. 4).

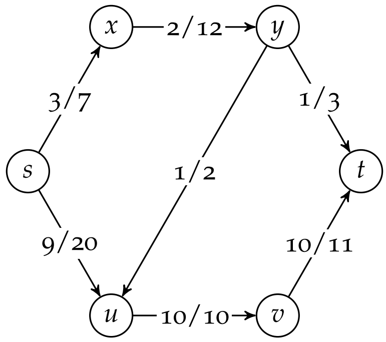

Figure 1: Flow network for problem 16.

(10 p) 17. Your friend Lurvik has invented an algorithm for finding spanning tries in connected, unweighted, undirected graphs. She starts with an arbitrary edge. She then goes through all the edges repeatedly. In each iteration she adds edges that have exactly one neighboring edge in the spanning tree. (The edges must be added individually. For each edge added one must also take into consideration which edges were added earlier in the same iteration.) Is the algorithm correct? Explain very briefly. If the graph has n nodes and m edges, what is the running time, in the worst case?

(10 p) 18. Assume that the classes P and NP are already defined (i.e., you need not define them.) Very briefly sketch the definition of NPC.

(10 p) 19. You have a row $v_1, \ldots, v_n$ of coins, where n is an even number. You and an opponent are to alternate taking either the first or the last coin of those that are left. Describe an algorithm that determines how much you are *guaranteed* to win, if you start.

(10 p) 20. You are given a graph where ach node is *red* or *green*, and there are no edges between nodes of the same color. A green node has 0, 1 or 2 neighbors, while the red ones may have arbitrarily many. Each green node also has a positive weight. You are to delete green nodes, but the resulting graph must still have paths between all red nodes. You wish to delete a set with as high total weight as possible. Describe an algorithm that solves the problem, and explain briefly why your answer is correct.