Umeå University
Department of Computing Science
5DV022 — Analysis of Algorithms
Examination: November 1, 2007

Name (printed): ______________________________

Swedish ID number: ______________________________

Computer User-ID: ______________________________

Signature: ______________________________

1. All answers must be written in English.
2. A pocket engineering calculator (miniräknare) and English/X - X/English dictionary may be used. No other help materials are allowed.
3. You must turn in this cover sheet together with the pages containing your answers. Collate the sheets in numerical order of the problems, and write on only one side of the paper.
4. The examination has a total of 400 points, which constitutes 40% of the course grade. The other 60% is based upon the obligatory exercises and problem sets, as explained in the course syllabus.
5. As noted in the course syllabus, it is necessary to obtain at least 100 points on the examination in order to receive a passing grade, regardless of the number of points obtained on the laboratory exercises and problem assignments.
6. In many of the problem with multiple parts, the answer to later parts depends upon the answer to earlier parts. If the answer to an earlier part is wrong, and if that error makes the answer to a later part much easier or trivial, then full credit will not be awarded for that later part even if the answer is correct relative to the previous error.
7. This examination consists of eight (8) problems on a total of four (4) pages.
8. You have the option of skipping a problem for partial credit. In the table below, place an X in the position for any problem for which you have attempted a solution, and which you wish to have graded. It is extremely important that you fill in this table properly, because of the following option. For any box which is left blank, the associated question will not be graded, and you will instead be awarded 12% of the points for that question. Your decision to leave a box blank is definitive, so be very careful. For example, If you leave box 5 blank, your answer to that question will not be graded, even if it is completely correct. On the other hand, if you place an X in box 5, but provide no answer whatsoever to that question, you will not receive 12% of the points for that question. It is strongly recommended that you use a pencil, in case you change your mind! Note that it is not possible to opt in or out of parts of problems; only entire problems may be skipped for partial credit.

Prob	1	2	3	4	5	6	7	8

(1: 50 points total) Two distinct notions of asymptotic complexity were defined in the course notes and lectures, each recapturing the notion that function f is at least as complex as function g. They were represented as $f \in \Omega_1(g)$ and $f \in \Omega_2(g)$. Define each of Ω_1 and Ω_2, and briefly discuss the advantages and disadvantages of each. (It is *not* necessary that you remember which was called Ω_1 and which was called Ω_2. Just define and compare the two notions.) For full credit, your answer must include formal mathematical definitions, as well as less formal explanations of their respective meanings. Illustrate each with a simple example.

(2: 50 points total) The worst-case complexity of an algorithm is characterized by the following recurrence relation

$$T(n) = 8 \cdot T(n/2) + 2 \cdot n$$

with initial condition $T(1) = 1$.

(a) Using a change of variable, obtain a linear recurrence relation.

(b) Give the characteristic polynomial of this linear recurrence.

(c) Solve for the roots of this polynomial.

(d) Give the solution to this linear recurrence.

(e) Translate this solution back to the original variable. For full credit, this answer must be in a simplified form.

(f) Give a tight bound on the worst-case complexity of the algorithm based upon this solution, in $\Theta(-)$ format.

(3: 50 points total) Answer the following questions. For (a)-(c), you are to give **formal** definitions, not "intuitive" ones. Furthermore, for parts (a) and (b), illustrate the concepts using the discrete knapsack problem.

(a) Define *subset system* and *profit function*.

(b) Define *combinatorial optimization problem*, *feasible solution*, and *optimal solution*, with respect to your answer of (a).

(c) Define the *formal greedy algorithm* with respect to a combinatorial optimization problem.

(d) Using the notions of (a)-(c), define the concept of *matroid*.

(e) Give two further conditions which are equivalent to a subset system being a matroid. There is no need to prove the equivalence; simply state the conditions.

(4: 50 points total) Consider the following variation on the job scheduling problem. Each job job_j has associated with it a profit $p(j)$, as in the problem in the course notes. However, each job now consists of two parts, each with its own deadline (a nonnegative integer), and each with unit running time. Thus, two deadlines, $d_1(\mathsf{job}_j)$ and $d_2(\mathsf{job}_j)$, are associated with each job job_j. A feasible solution S is one in which both parts of each job in S may be scheduled to complete before their respective deadlines. The profit associated with a solution is the sum of the individual profits. An optimal solution is a feasible one with maximal profit. Prove or disprove: this problem defines a matroid.

Note: A statement such as "It is a matroid" or "It is not a matroid", without a correct proof, will not receive credit.

(5: 50 points total) Consider the following statement:

No sorting algorithm can have average-case complexity which is better than $O(n \cdot \log(n))$.

Sketch an argument which shows this to be the case. Make sure that you state the assumptions under which this argument holds.

(6: 50 points total) Answer the following questions in the context of the discrete knapsack problem P with n objects $\{\mathsf{obj}_1, \mathsf{obj}_2, \ldots, \mathsf{obj}_n\}$, each object obj_i possessing a weight w_i and a value v_i. The capacity of the knapsack is M.

(a) Given $\varepsilon > 0$, define what is meant by an ε-relative polynomial-time approximation algorithm for P.

(b) In the course, a number of ε-relative polynomial-time approximation algorithms for P were presented. Sketch one of these algorithms, using high-level pseudocode, in sufficient detail that a competent programmer would have no difficulty filling in the details.

(c) Give the worst-case complexity of the algorithm given in part (b), as a function of ε and n. (It is quite possible that ε itself will be specified in terms of another parameter, say k. If so, also indicate the relationship between ε and k.) No proof is required; a statement of the theorem will suffice.

(7: 50 points total) In Laboratory Exercise 3, it was generally observed that, over the range of test data used, the worst-case performance of building a min-heap by repeated insertion was approximately linear in the number n of input elements, rather than $\Theta(n \cdot \log(n))$ as predicted by mathematical analysis. Discuss the reasons which might lie behind this apparent contradiction, and how a further investigation might be crafted which would provide a more meaningful analysis.

(8: 50 points total) Given is the following matrix representation for a directed graph.

$$\begin{bmatrix} \infty & 7 & 3 & 12 & 8 \\ 3 & \infty & 6 & 14 & 9 \\ 5 & 8 & \infty & 6 & 18 \\ 9 & 3 & 5 & \infty & 11 \\ 18 & 14 & 9 & 8 & \infty \end{bmatrix}$$

Shown on the next page is the first level of expansion of the travelling-salesman defined by this graph, using the branch-and-bound algorithm described in the course notes and the textbook. Extend this partial solution to a complete, optimal one. In your solution, list the vertices of the corresponding search tree in order of generation. For each such vertex, give the following.

(a) The index of its generation.

(b) The path in the graph to which the vertex corresponds.

(c) The value $\hat{c}$ of the node, expressed as a sum of the value of its parent, the cost of the edge corresponding to travelling from its parent, the amount added by row reduction, and the amount added by column reduction. (Rows must be reduced before columns.) In the case that it is a near-leaf, add in the cost of the final path back to the start vertex as well. (For the root node, there will not be any row and column reduction.)

(d) Four versions of the associated cost matrix.

(i) before any reduction whatever;

(ii) before any row or column reduction, but after the assignment of certain entries to the value of ∞ from dynamic reduction (not applicable to the root vertex);

(iii) after row reduction but before column reduction, with the value by which each row was reduced;

(iv) after both row and column reduction, with the value by which each column was reduced.

Make sure that you identify explicitly the final optimal solution, as a path through the graph. Only those vertices actually generated in the branch-and-bound search need be listed. To receive credit, your answers must be in the format described above and illustrated on the next page.

Vertex 1; path 1; $\hat{c} = 22 + 5 = 27$:

$$\begin{bmatrix} \infty & 7 & 3 & 12 & 8 \\ 3 & \infty & 6 & 14 & 9 \\ 5 & 8 & \infty & 6 & 18 \\ 9 & 3 & 5 & \infty & 11 \\ 18 & 14 & 9 & 8 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & 4 & 0 & 9 & 5 \\ 0 & \infty & 3 & 11 & 6 \\ 0 & 3 & \infty & 1 & 13 \\ 6 & 0 & 2 & \infty & 8 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \begin{array}{c} 3 \\ 3 \\ 5 \\ 3 \\ 8 \\ \hline 22 \end{array} \mapsto \begin{bmatrix} \infty & 4 & 0 & 9 & 0 \\ 0 & \infty & 3 & 11 & 1 \\ 0 & 3 & \infty & 1 & 8 \\ 6 & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 0 \quad 0 \quad 5 \quad - \quad 5$$

Vertex 2; path 12; $\hat{c} = 27 + 4 + 3 + 0 = 34$:

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 0 \\ 0 & \infty & 3 & 11 & 1 \\ 0 & 3 & \infty & 1 & 8 \\ 6 & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 3 & 11 & 1 \\ 0 & \infty & \infty & 1 & 8 \\ 6 & \infty & 2 & \infty & 3 \\ 10 & \infty & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 2 & 10 & 0 \\ 0 & \infty & \infty & 1 & 8 \\ 4 & \infty & 0 & \infty & 1 \\ 10 & \infty & 1 & 0 & \infty \end{bmatrix} \begin{array}{c} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ \hline 3 \end{array} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 2 & 10 & 0 \\ 0 & \infty & \infty & 1 & 8 \\ 4 & \infty & 0 & \infty & 1 \\ 10 & \infty & 1 & 0 & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 0 \quad 0 \quad 0 \quad - \quad 0$$

Vertex 3; path 13; $\hat{c} = 27 + 0 + 1 + 1 = 29$:

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 0 \\ 0 & \infty & 3 & 11 & 1 \\ 0 & 3 & \infty & 1 & 8 \\ 6 & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 11 & 1 \\ \infty & 3 & \infty & 1 & 8 \\ 6 & 0 & \infty & \infty & 3 \\ 10 & 6 & \infty & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 11 & 1 \\ \infty & 2 & \infty & 0 & 7 \\ 6 & 0 & \infty & \infty & 3 \\ 10 & 6 & \infty & 0 & \infty \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \hline 1 \end{array} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 11 & 0 \\ \infty & 2 & \infty & 0 & 6 \\ 6 & 0 & \infty & \infty & 2 \\ 10 & 6 & \infty & 0 & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 0 \quad 0 \quad 1 \quad - \quad 1$$

Vertex 4; path 14; $\hat{c} = 27 + 9 + 1 + 1 = 38$:

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 0 \\ 0 & \infty & 3 & 11 & 1 \\ 0 & 3 & \infty & 1 & 8 \\ 6 & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & \infty & 1 \\ 0 & 3 & \infty & \infty & 8 \\ \infty & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & \infty & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & \infty & 1 \\ 0 & 3 & \infty & \infty & 8 \\ \infty & 0 & 2 & \infty & 3 \\ 9 & 5 & 0 & \infty & \infty \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \hline 1 \end{array} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & \infty & 0 \\ 0 & 3 & \infty & \infty & 7 \\ \infty & 0 & 2 & \infty & 2 \\ 9 & 5 & 0 & \infty & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 0 \quad 0 \quad 1 \quad - \quad 1$$

Vertex 5; path 15; $\hat{c} = 27 + 0 + 0 + 1 = 28$:

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 0 \\ 0 & \infty & 3 & 11 & 1 \\ 0 & 3 & \infty & 1 & 8 \\ 6 & 0 & 2 & \infty & 3 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & 11 & \infty \\ 0 & 3 & \infty & 1 & \infty \\ 6 & 0 & 2 & \infty & \infty \\ \infty & 6 & 1 & 0 & \infty \end{bmatrix} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & 11 & \infty \\ 0 & 3 & \infty & 1 & \infty \\ 6 & 0 & 2 & \infty & \infty \\ \infty & 6 & 1 & 0 & \infty \end{bmatrix} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \hline 0 \end{array} \mapsto \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 2 & 11 & \infty \\ 0 & 3 & \infty & 1 & \infty \\ 6 & 0 & 1 & \infty & \infty \\ \infty & 6 & 0 & 0 & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 1 \quad 0 \quad 0 \quad - \quad 1$$