Norwegian University of Science and Technology
Department of Computer and Information Science

Final Exam in TDT4125 Algorithm Construction

Read the entire exam before you start, plan your time, and prepare any questions for when the teacher comes to the exam room. Make assumptions where necessary. Keep your answers short and concise. Long explanations that do not directly answer the questions are given no weight.

Points are given for each subproblem. These sum to 120, but the maximum score is 100 points. In other words: **A score of more than 100 counts as 100.**

Do not waste your time on extensive explanations.

The most important thing is to show that you have understood the main points. Usually, a sentence or two will suffice.

Problem 1

a) If we are unable to find the optimum for a problem, how is it possible to guarantee that we are within a given factor? Explain briefly. (6 p)

b) If you express a problem as a linear program with integer constraints, and then produce a new linear program that differs from the original only in that you have removed the integer constraints, what can you say about the relationship between the two programs? (6 p)

c) What is the difference between a *polynomial-time approximation scheme* (PTAS) and a *fully polynomial-time approximation scheme* (FPTAS)? Explain briefly. (6 p)

d) If (x, k) is an instance of a parameterized problem, how do we define the *size* of this instance? (6 p)

e) If you have an instance (G, k) of the problem VERTEX COVER, why is it safe to reduce this to $(G - v, k - 1)$ if v has at least $k + 1$ neighbors? (6 p)

f) What are the two reductions (FAST.1 og FAST.2) for the problem FEEDBACK ARC SET IN TOURNAMENTS? (6 p)

g) Explain briefly why T_P for a greedy scheduler is at most a factor 2 away from the time of an optimal scheduler. (6 p)

h) Explain briefly how to simulate CRCW using EREW. Describe how both reading and writing is handled. (6 p)

i) Assume that you have a weighted, directed graph $G = (V, E)$ with a given start node s and an end node t. You want to find a path p from s to t such that the greatest edge weight in p is as small as possible. Express this problem as a linear program without integer constraints. (6 p)

j) For a qubit $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, what are the probabilities of outcomes $|0\rangle$ og $|1\rangle$? (6 p)

k) What kind of matrices are used to represent quantum gates? (6 p)

Problem 2

Your friend Lurvik has written a semester report on "the absence of cliques" in a graph. The text alternates between two interpretations of this description: Either that there is an absent clique (that is, an independent set) of size k or greater in the graph, or that there is no clique of size k or greater in the graph. Consider these two interpretations as decision problems.

a) You point out that these two problems are not identical, but Lurvik believes they are equivalent, i.e., that they may be reduced to each other in polynomial time. Do you agree? Discuss briefly. (7 p)

Lurvik is now trying to find a *maximum cut* in an undirected graph, that is, a partitioning of its nodes into two sets A and B such that the number of edges between A and B is as great as possible.

She has taken a course on heuristic optimization, and she has now constructed an algorithm based on so-called *local search* or *hill-climbing*, where a solution is gradually improved until you can't get any further. The basic operation is to *move* a node, that is, to let it shift from A to B or from B to A. We say that a node *should be moved* if moving it results in a better solution. Her algorithm works as follows:

```
HEURISTIC-MAX-CUT(G)
1  construct a random partitioning A, B of G.V
2  while there is a v ∈ G.V such that v should be moved
3        move v
4  return A, B
```

Algorithms based on local search generally have several drawbacks. For example, it is not a given that they will terminate in polynomial time, and they may get stuck in local optima, thereby not solving the problem optimally. Even so, you think there is something intriguing about Lurvik's algorithm.

b) Lurvik thinks the algorithm is sure to find a *pretty good* solution in reasonable time, for example that it is guaranteed to achieve a certain percentage of the optimum, and that it will terminate in polynomial time. Do you agree? Discuss briefly. (7 p)

c) Can you say something similar about the algorithm if you delete lines 2 and 3? State any assumptions. Discuss briefly. (6 p)

d) If you were to start with the algorithm in subproblem c to construct a deterministic algorithm, how would you proceed? Explain briefly. (6 p)

Problem 3

You are to place emergency vehicles so as to minimize response times. From a set L of n locations you are to designate a set $S \subseteq L$ of m locations as *stations*, each of which receives a vehicle. The remaining locations are *destinations*. For each pair of locations $i, j \in L$ you are given the response time t_{ij} from i to j in minutes, as an integer. You may assume that $t_{ij} = t_{ji}$ and that all shortcuts are used, so that the *triangle inequality* holds:

$$t_{ik} \leq t_{ij} + t_{jk} \qquad \forall i, j, k \in L\,.$$

During an emergency, we assume that it is (uniformly) random which station responds. This means that each destination $j \notin S$ has an *expected response time* that is the average of the response times from the m stations, that is,

$$e_j = \frac{1}{m} \sum_{i \in S} t_{ij} \,.$$

You are tasked with giving guarantees for these expected response times, and therefore wish to select S such that you minimize the greatest expected response time $\max_{j \in L \setminus S} e_j$.

Rather than simply solving this one instance, you wish to construct an algorithm that solves the problem in general. Where it is natural to parameterize the problem, use m as the parameter.

a) Show that the problem is in XP (*slice-wise polynomial*). (7 p)

b) Show that the problem is W[2]-hard. (7 p)

c) Show that the problem does not have a PTAS, unless P = NP. (7 p)

d) Describe an approximation algorithm for the problem, with $\alpha = 2$. What is the running time? (7 p)

Hint: Show $t_{jk} \leq e_j + e_k$. For partial credit, simply assume this.