NTNU
Institutt for datateknikk
og informasjonsvitenskap

Examination paper for TDT4120 Algorithms and Data Structures

Academic contact during examination	Magnus Lie Hetland
Phone	918 51 949
Examination date	Dec 12, 2014
Examination time (from–to)	0900–1300
Permitted examination support material	D. No printed/handwritten materials permitted. Specific, simple calculator permitted.
Language	English
Number of pages	9
Number of pages enclosed	0
Checked by	Pål Sætrom

_____

Please read the entire exam carefully before you start. Plan your time, and prepare any questions for when the teacher comes to the exam room.

* * *

Make assumptions where necessary. Write your answers where indicated, and keep them short and concise, if possible. Long explanations that do not directly answer the questions are given little or no weight.

Unless a given problem states otherwise, you may describe your algorithms in prose, pseudocode or program code, according to preference, as long as it is clear how the algorithm works. Short, abstract explanations may be just as good as extensive pseudocode, as long as they are clear and precise.

Your algorithms should be as efficient as possible, unless otherwise stated. Running times are to be given in asymptotic notation, as precisely as possible.

The maximum score attainable is indicated for each problem. The maximum total score attainable is 100 points.

Note: In Problem 2b, the notation $v.d$ follows the most recent edition of the textbook. This is equivalent to the notation $d[v]$ used in previous editions.

Problem 1

a) Which problem does Dijkstra's algorithm solve?

Answer (5 pts):

b) Which traversal algorithm do we use for topological sorting?

Answer (5 pts):

c) Which design method would you use to design an algorithm to fully parenthesize a matrix-chain multiplication?

Answer (5 pts):

d) What does Prim's algorithm keep in its min-priority queue?

Answer (5 pts):

e) What is the typical average-case running time for sorting algorithm based on the divide-and-conquer design method?

Answer (5 pts):

Problem 2

a) Which design method would you use in designing an algorithm for the fractional knapsack problem?

Answer (5 pts):

b) During the execution of some shortest-path algorithm, $u.d = 5$, $v.d = 7$, $w(u, v) = 1$ and $w(v, u) = -2$. After one call to RELAX(u, v, w) and a subsequent call to RELAX(v, u, w), what is the value of $u.d$?

Answer (5 pts):

c) If $a \cdot \varphi(i) \leq b \cdot \psi(i)$ for every $i \geq k$, what can you say about the asymptotic relationship between φ and ψ? (Here φ and ψ are non-negative real-valued functions, defined on the positive integers, a and b are positive real-valued constants and k is a positive integer constant.)

Answer (5 pts):

d) Very briefly, what characterizes a good hash function?

Answer (5 pts):

e) You are examining an unfamiliar problem A, and you wish to relate it to problem B, which you know has a running time of $\Omega(n^2)$ in the worst case. You wish to show that the same bound holds for A, using a reduction. Very briefly, what can you say about this reduction?

Answer (5 pts):

Problem 3

You wish to sort the sequence $A = (a_1, a_2, \ldots, a_n)$. It is well-known that for comparison-based sorting, the best running time achievable, in the expected and worst cases, is $\Omega(n \log n)$.

a) Assume that the elements are real numbers, distributed according some given probability distribution, which can be computed in constant time for any element. What is the best running time you could get, in the average case, and how would you get it? Explain your reasoning briefly.

Answer (5 pts):

b) Now assume that the elements are positive integers, and that $a_i \leq p(n)$ for $i = 1 \ldots n$, where p is some polynomial. What is the best running time you could get, in the worst case, and how would you get it? Explain your reasoning briefly.

Answer (5 pts):

Problem 4

a) Which problem does Floyd-Warshall solve? What assumption must you or can you make, and how do they affect the running time? Explain briefly.

Answer (5 pts):

b) Which problem does Ford-Fulkerson solve? What assumption must you or can you make, and how do they affect the running time? Explain briefly.

Answer (5 pts):

Problem 5

a) Solve the recurrence $T(n) = 2T(n/2) + T(n)/2 + n$, where $T(1) = 1$. Give your answer in asymptotic notation.

Answer (5 pts):

b) An algorithm processes an image in the form of an $n \times n$ matrix of pixels, by splitting it up into non-overlapping squares of size $(n/2) \times (n/2)$, processing these recursively, and then combining the results in linear time, as the function of the number of pixels involved. What is the total running time, as a function of n?

Answer (5 pts):

Problem 6

You are planning to plant gardens and dig irrigation canals from some given water source. You can decide the layout yourself, and have decided on the following. Starting at the source, the canals form a binary tree structure, with the gardens acting as leaves. The distancing is completely regular. The distance along a canal from the source to a fork, or from a fork to a leaf or another fork is the same, and is set to $\ell(n)$ for a given number of gardens, n.

a) What is the total length of canal that must be dug, as a function of n?

Answer (5 pts):

Each garden i requires an amount of water equal to w_i per day. To accommodate this, some of the canal segments will need to be wider than others, leading to more digging. If garden i is at a distance of d_i segments from the water source, its contribution to the digging needed is proportional to $d_i w_i$. The total amount of digging will thus be proportional to $\sum_{i=1}^{n} d_i w_i$.

b) Describe an algorithm for deciding how to construct a tree of canals so that the total amount of digging is minimized.

Answer (5 pts):

Problem 7

Consider the *balanced partition problem*, where we are given n integers $a_1, \ldots, a_n$ in the range $0 \ldots k$, and we want to partition them into two sets S_1 and S_2 so that $\left| \sum_{x \in S_1} x - \sum_{y \in S_2} y \right|$ is minimized.

a) Show that the problem is NP-hard.

Answer (5 pts):

b) Sketch an algorithm for solving the problem in pseudopolynomial time.

Answer (5 pts):