Final exam in
TDT4120 Algorithms og data structures

Exam date	14 December 2011
Exam time	1500–1900
Grading date	14 Januar
Language	English
Concact during the exam	Magnus Lie Hetland (ph. 91851949)
Tillatte Aids	No printed/handwritten; specific, simple calculator

! **Please 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 little or no weight.

You may describe your algorithms in text, pseudocode or program code, as long as it is clear how the algorithm works. Short, abstract explanations can be just as good as extensive pseudocode, as long as they are precise enough. Running times are to be given in asymptotic notation, as precisely as possible.

Problem 1 (44%)

a) Assume that the nodes of a directed graph can be ordered from left to right so that all edges point from left to right. What is such an ordering called?

Answer (4%):

b) Assume that the undirected graph $G = (V, E)$ is connected. What is the name of the smallest subset F of E such that (V, F) is connected? (Note: There may be more than one such subset.)

Answer (4%):

c) Which design principle (dynamic programming, divide and conquer eller greed) is used in Huffman's algorithm?

Answer (4%):

d) What is the running time of QUICKSORT in the worst case?

Answer (4%):

e) What kind of queue is used in DFS?

Answer (4%):

f) Which algorithm would you use to find the shortest path between two nodes in an undirected, unweighted graph?

Answer (4%):

g) To find the minimum cut (min-cut) is equivalent to another curriculum problem. Which one?

Answer (5%):

h) You know that problem A is in NP and problem B is in NPC. You wish to show that A is also in NPC. Which way would you reduce?

Answer (5%):

i) Which algorithm would you use to find the shortest path between two nodes in a directed, weighted graph if you cannot make any assumptions about the edge weights?

Answer (5%):

Assume that you are dividing players in some ball game into two teams. You have estimated the skill of each player as a positive real number, and you wish to distribute the players as *unevenly* as possible (that is, you want the sums of the skill values for the two teams to be as different as possible). The teams should have an equal number of players.

j) How would you solve the division problem with running time $\Theta(n \lg n)$?

Answer (5%):

Problem 2 (35%)

a) You are constructing a complete, directed graph with nodes $1, 2, \dots, n$. If you can freely choose the direction of every edge, how many different graphs can you construct?

Answer (7%):

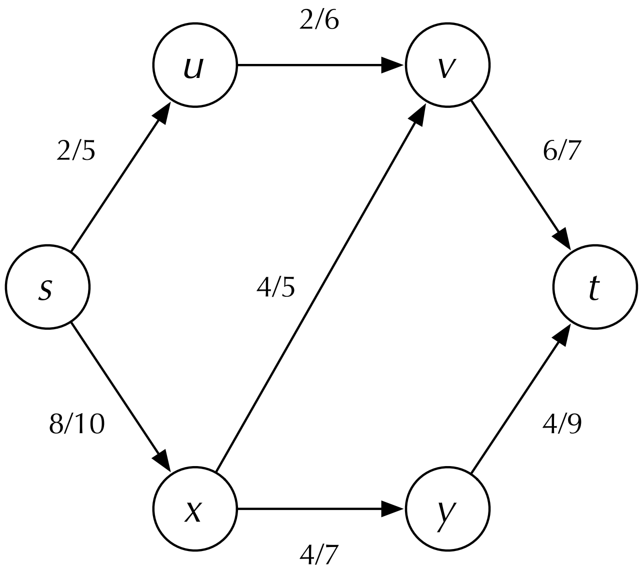
b) You have two sets **A** and **B** (both of size n) and a number x. Describe an efficient algorithm that lets you decide if there is a number y in **A** and another number z in **B** such that $x = y + z$. What is the running time?

Answer (7%):

c) A friend of yours claims that he has developed a general priority queue where the operations for adding an element, finding the maximum, and removing the maxumum each has a running time of $O(1)$ in the worst case. Explain why this cannot be true.

Answer (7%):

Consider the following flow network over the nodes $\{s, t, u, v, x, y\}$, with source s and sink t:



Flow and capacity are given by the edges (for example, the flow from x to y is 4, with a capacity of 7).

d) Give the augmenting path that yields the highest flow increase. Give your answer as a sequence of nodes.

Answer (7%):

e) Give (e.g., draw) a (small) example of a shortest path tree (for example as produced by Dijkstra's algorithm) that is not a minimum spanning tree.

Answer (7%):

Problem 3 (21%)

Let T be an unweighted, undirected acyclic graph (that is, a tree without a specified root). Let the *diameter* of T be the length of (the number of edges in) the longest path in T.

a) Describe an efficient algorithm to compute the tree diameter as described. What is the running time?

Answer (7%):

In a tree T with root r a node v is a *descendant* of u if (and only if) there is a path from r to v that passes through u. You now wish, for a given pair of nodes u and v, to determine if u is a descendant of v, if v is a descendant of u, or if neither is a descendant of the other. We can call this the *descendant problem*.

b) Describe an algorithm that preprocesses the tree T with a running time of $O(n)$ so that the descendant problem can be solved for T with a running time $O(1)$.

Answer (7%):

You are a theater director, and you are making a rehearsal schedule – a schedule indicating which actors need to attend on which days during the rehearsal of a play. You have the following information:

- A set of actors, where actor *i* is paid to attend *D*[*i*] rehearsal days (and thus cannot attend any more days than this).
- A set of scenes, where scene *i* is to be rehearsed on *S*[*i*] of the rehearsal days. The sum of *S*[*i*] over all the scenes is equal to the number of days available. (Only one scene is rehearsed on any given day.)
- Which actors are available on which days. (Some of them are busy on some of the rehearsal days.)
- Which actors are in which scenes (and therefore must attend if the scene is to be rehearsed).

You wish to develop an algorithm that constructs a rehearsal schedule, that is, that descides which scene is to be rehearsed on each of the rehearsal days. If such a schedule is cannot be constructed, the algorithm is to report this.

c) Briefly describe an algorithm that solves the problem. (Feel free to illustrate your answer.)

Answer (7%):