Technical University of Denmark

Written exam, December 12, 2019.

Course name: Algorithms and data structures II.

Course number: 02110.

Aids allowed: Written aids are permitted.

Exam duration: 4 hours

Weighting: Question 1: 32% - Question 2: 15% - Question 3: 7% - Question 4: 20% - Question 5: 26%

The weighting is only an approximative weighting.

You can answer the exam in either Danish or English.

All questions should be answered by filling out the box below the question.

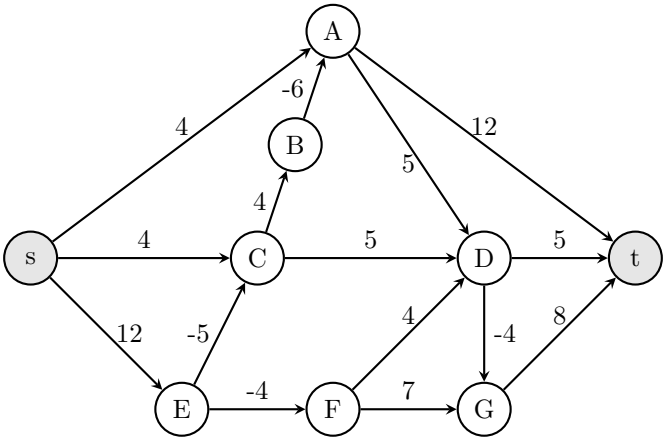
As exam paper just hand in this and the following pages filled out.

If you need more space you can use extra paper that you hand in together with the exam paper. If you do this, then indicate it in the exam set in the solution box for the relevant question.

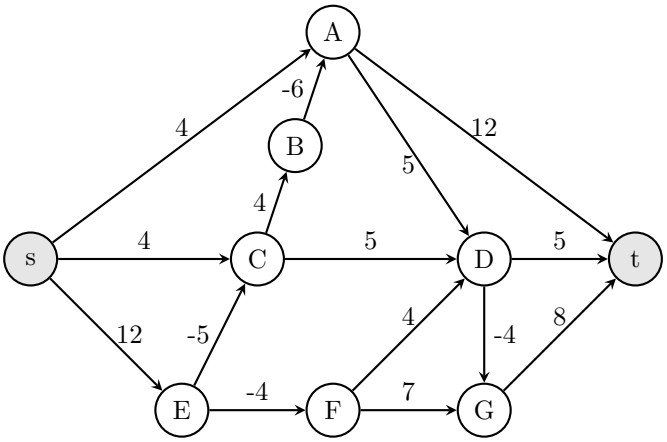
Study number: _____ Name: _____

Question 1

Question 1.1 (8%) Find a shortest path from s to t in the graph below. Mark the edges on the shortest path and write the length of the path.



Solution:



Length of shortest path: _____

Question 1.2 (8%) Compute the Fenwick tree F for the array A below:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$A[i]$	–	5	3	6	1	1	3	2	4	4	5	2	5	7	6	5	3

Solution:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$F(i)$																	

Question 1.3 (8%) Draw the string-matching finite automaton for the string `ALALGALA`:

Solution:

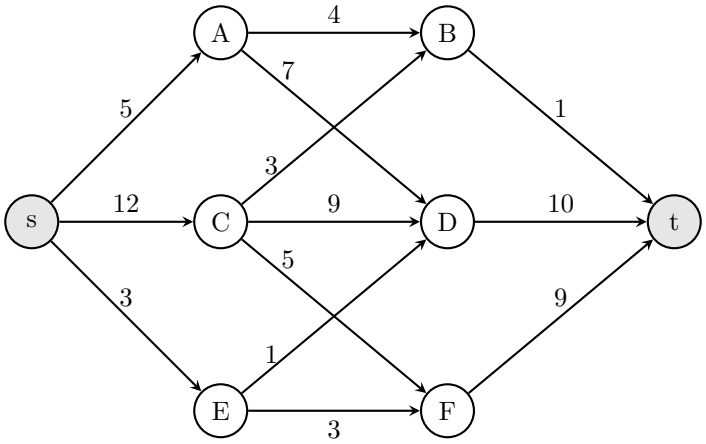
Question 1.4 (8%) Compute the prefix function as used in the Knuth-Morris-Pratt algorithm for the string `ALALGALA`:

Solution:

i	1	2	3	4	5	6	7	8
$P[i]$	A	L	A	L	G	A	L	A
$\pi[i]$								

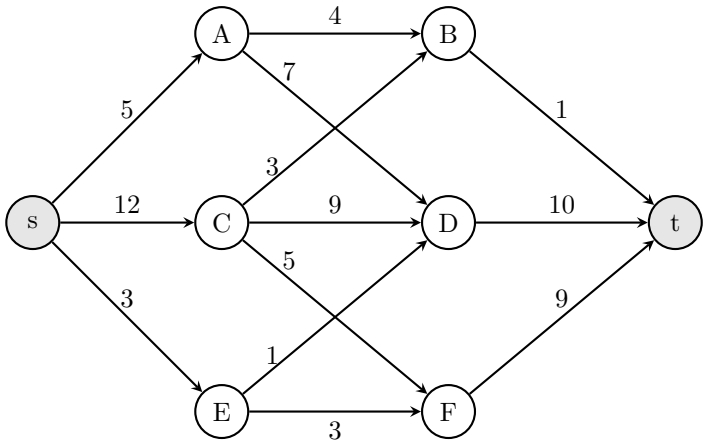
Question 2

Consider the network N below with capacities on the edges.



Question 2.1 (8%) Give a maximum flow from s to t in the network N (write the flow for each edge along the edges on the graph below), give the value of the maximum flow, and give a minimum $s-t$ cut (give the partition of the vertices).

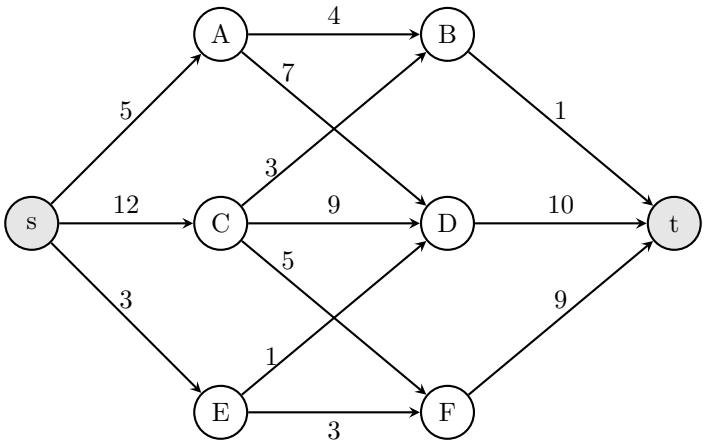
Solution:



value of flow: _____

minimum cut: _____

Question 2.2 (7%) Use the *Scaling Max-Flow Algorithm* to compute a maximum flow in the network N (the network is shown again below). For each augmenting path write the nodes on the path and the value you augment the path with in the table below.



Solution:

augmenting path	value

Question 3

Consider the following recurrence: $T(n) = T(n/2) + 4T(n/4) + cn^2$. Solve the recurrence using either a recursion tree or the substitution method.

Solution:

Solution to recurrence: _____

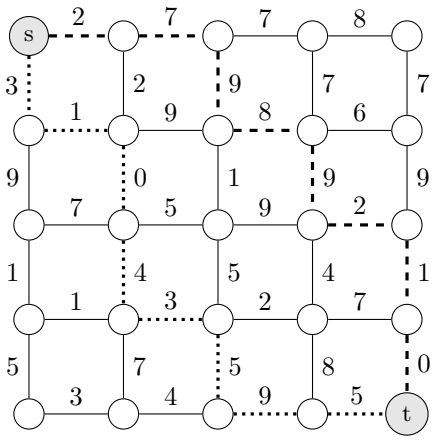
Argumentation/proof:

Question 4

Your are helping the tourist guide company "Manhattan Tourists", that are arranging guided tours of the city. They want to find a walk between two points on the map that is both interesting and short. The map is a square grid graph. The square grid graph has n rows with n nodes in each row. Let node $v_{i,j}$ denote the jth node on row i. For $1 \leq i < n$ and for $1 \leq j \leq n$ node $v_{i,j}$ is connected to $v_{i+1,j}$. And for $1 \leq i \leq n$ and for $1 \leq j < n$ node $v_{i,j}$ is connected to $v_{i,j+1}$. The edges have non-negative edge weights that indicate how interesting that street is. See the graph below for an example of a 5×5 grid graph.

They want to find a short interesting walk from the upper left corner ($s = v_{1,1}$) to the lower right corner ($t = v_{n,n}$). More precisely, they want to find a path with the possible smallest number of edges, and among all paths with this number of edges they want the path with the maximum weight (the weight of a path is the sum of weights on the path).

All shortest paths have $2n - 2$ edges and go from s to t by walking either down or right in each step. In the example below two possible shortest paths (of length 8) are indicated. The dashed path has weigth 38 and the dotted path has weight 30.



Let $W[i, j]$ be the maximal weight you can get when walking from s to $v_{i,j}$ walking either down or right in each step. Let $D[i, j]$ be the weight of the edge going down from $v_{i,j}$ and let $R[i, j]$ be the weight of the edge going right from $v_{i,j}$.

Question 4.1 (5%) Fill out the table for the example above:

$W_{i,j}$	1	2	3	4	5
1					
2					
3					
4					
5					

Question 4.2 (5%) Which of the following recurrences correctly computes $W[i, j]$:

A
$$W[i,j] = \begin{cases} 0 & \text{if } i = 1 \text{ and } j = 1 \\ W[i-1,1] + D[i-1,1] & \text{if } i > 1 \text{ and } j = 1 \\ W[1,j-1] + R[1,j-1] & \text{if } i = 1 \text{ and } j > 1 \\ W[i-1,j-1] + \max\{D[i-1,j-1] + R[i,j-1], R[i-1,j-1] + D[i-1,j]\} & \text{otherwise} \end{cases}$$

B
$$W[i,j] = \begin{cases} 0 & \text{if } i = 1 \text{ and } j = 1 \\ W[i-1,1] + D[i-1,1] & \text{if } i > 1 \text{ and } j = 1 \\ W[1,j-1] + R[1,j-1] & \text{if } i = 1 \text{ and } j > 1 \\ \max\{W[i,j-1] + D[i,j-1], W[i-1,j] + R[i-1,j]\} & \text{otherwise} \end{cases}$$

C
$$W[i,j] = \begin{cases} 0 & \text{if } i = 1 \text{ and } j = 1 \\ W[i-1,1] + D[i-1,1] & \text{if } i > 1 \text{ and } j = 1 \\ W[1,j-1] + R[1,j-1] & \text{if } i = 1 \text{ and } j > 1 \\ \max\{W[i-1,j] + D[i-1,j], W[i,j-1] + R[i,j-1]\} & \text{otherwise} \end{cases}$$

Question 4.3 (10%) Write pseudocode for an algorithm *based on dynamic programming and the recurrence from Question 4.2* that computes the maximum weight a shortest path can have in the grid graph.

Analyze the space usage and running time of your algorithm in terms of n.

Solution:

Solution 4.3 continued:

Question 5

The tourist guide company "Manhattan Tourists" are arranging several tours of the city at the same time. As in Question 4 the map is a square grid graph, but now some of the streets are blocked, so they can't walk on these. They want to arrange several tours from s to t that do not overlap.

Question 5.1 (10%)

The company wants to arrange two different tours from s to t that are disjoint in the sense that they don't share any streets. In this question you don't care whether the tours are interesting (the weight on the path does not matter). The tours should still only walk down and right in each step and you cannot walk on the blocked streets. Give an efficient algorithm that determines wether it is possible to make two such tours. Analyze the asymptotic running time of your algorithm in terms of n. Remember to argue that your algorithm is correct.

Solution:

Solution 5.1 continued:

Question 5.2 (8%)

It turned out that the tours were not different enough. The company now wants to arrange two different tours from s to t that are disjoint in the sense that the two tours only meet in s and t. In this question you don't care wether the tours are interesting (the weight on the path does not matter). The tours should still only walk down and right in each step and you cannot walk on the blocked streets. Give an algorithm that determines wether it is possible to make two such tours. Analyze the asymptotic running time of your algorithm in terms of n. Remember to argue that your algorithm is correct.

Solution:

Solution 5.2 continued:

Question 5.3 (8%)

The company would like their tours to be not only disjoint but also interesting. So they want to arrange two *street disjoint* tours where the least interesting street on the tours is as interesting as possible. That is, find the maximum value w such that there are two street disjoint tours that both uses only streets of weight w or higher. The tours should still only walk down and right in each step and you cannot walk on the blocked streets. The weights on the edges are integers between 1 and W. Give an efficient algorithm that solves the problem. Analyze the asymptotic running time of your algorithm in terms of n and W. Remember to argue that your algorithm is correct.

Solution:

Solution 5.3 continued: