Technical University of Denmark

Written exam, December 8, 2023.

Course name: Algorithms and data structures II.

Course number: 02110.

Aids allowed: Written aids are permitted.

Exam duration: 4 hours

Weighting: Question 1: 28% - Question 2: 8% - Question 3: 14% - Question 4: 12% - Question 5: 16% - Question 6: 22 %.

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 only hand in this and the following pages filled out.

Study number: ______________

Table number: ______________

The exam set continues on the next page

Question 1 (28%)

Question 1.1 (7%) Draw the string matching automaton for the string `DDADDBDD` (leave out the edges going back to the starting state):

Solution:

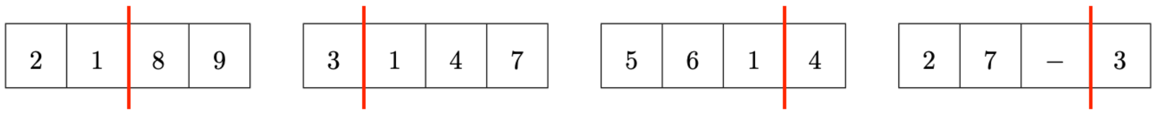
Question 1.2 (7%) Compute the prefix function as used in the Knuth-Morris-Pratt algorithm for the string `DDADDBDD`:

Solution:

i	1	2	3	4	5	6	7	8
$P[i]$	`D`	`D`	`A`	`D`	`D`	`B`	`D`	`D`
$\pi[i]$								

The exam set continues on the next page

Question 1.3 (7%) Consider the 2-level rotated array data structure below.



Question 1.3.1 (4%). Show the result of applying the operation Insert(3, 6). (Recall that Insert(i, x) inserts a new entry with value x immediately *to the left* of entry i.)

Solution:

Question 1.3.2 (3%). Show the result of applying the operation Delete(5) on the *original 2-level array.*

Solution:

Question 1.4 (7%) 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
$A[i]$	–	4	2	1	6	3	3	7	8	2	3	4	4	1	2	1	9

Solution:

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

Question 2 (8%)

Let $T(n)$ denote the running time of an algorithm and let $T(n)$ be defined by the following recurrence.

$$T(n) = \begin{cases} T(n/3) + T(n/6) + cn & \text{if } n > 5 \\ c & \text{otherwise} \end{cases}$$

Question 2.1 (5%) Draw the recursion tree for the recurrence. You should draw at least the first 4 levels. Write

- the size of the subproblem represented by each node,
- the time used for the subproblem represented by each node,
- and the total time used on each level drawn in the tree.

Recursion tree:

The exam set continues on the next page

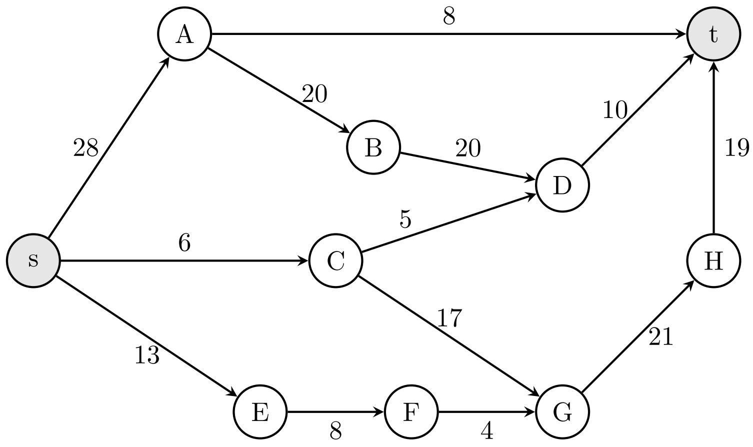
Question 2.2 (1%) The height of the recursion tree is at most: ________________

Question 2.3 (2%) The total time used on level j can be expressed as:

$\boxed{\text{A}}$ $\left(\frac{1}{6}\right)^j cn$ $\boxed{\text{B}}$ $\left(\frac{1}{3}\right)^j cn$ $\boxed{\text{C}}$ $\left(\frac{1}{2}\right)^j cn$ $\boxed{\text{D}}$ $\left(\frac{2}{3}\right)^j cn$ $\boxed{\text{E}}$ cn

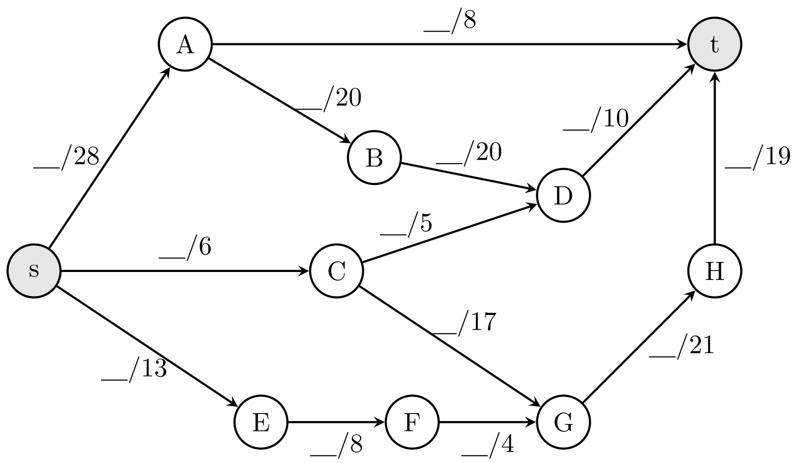
Question 3 (14%)

Consider the network N below with capacities on the edges.



Question 3.1 (7%) 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).

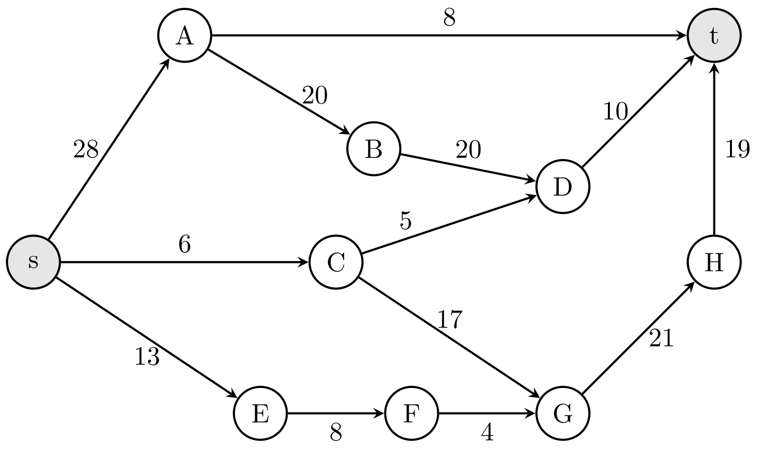
Solution:



value of flow: ____________________

minimum cut: ____________________

Question 3.2 (7%) Use the *Edmonds-Karp 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 4 (12%)

Your friends have a bookstore. They have n books $b_1, b_2, \ldots, b_n$. Initially, each book b_i has a price $p_i \in \{1, 2, \ldots, P\}$, where $P < n$. They regularly change the prices of the books and they ask you to help them implement a system that can perform the following operations:

- ReturnPrice(i): Return the price of book i.
- ChangePrice(i, p): Change the price of book i to p.
- Count(i,j): Return the number of books with a price between i and j (both included).

Help your friends by giving a data structure that efficiently supports the operations. As input to the data structure you are given the list of books $\{b_1, b_2, \ldots, b_n\}$ and the list of their prices $\{p_1, p_2, \ldots, p_n\}$. Remember to analyse the space usage and query time of your data structure and argue for the correctness.

Solution:

Solution 4 continued:

Question 5 (16%)

You want to read as many pages as possible before the next meeting in the book club. You know you have time to read P pages. You have B books and for each book i you know the number of pages p_i that it contains. There is a catch though. The pages of a book only count if you read the whole book.

Example: Let $P = 8$ and $B = 3$ where $b_1 = 3$, $b_2 = 4$ and $b_3 = 6$. Then the maximum number of pages you can obtain is 7 by reading books 1 and 2.

Question 5.1 (2%) Your friend suggests the following strategy: Start by sorting the books according to the number of pages and then read them in that order (the one with the fewest pages first).

Give an example that shows that the algorithm suggested by your friend does not find the optimal solution. You should write the instance, the optimal solution, and the solution your friends' algorithm would find.

Solution:

Question 5.2 (14%) Give an algorithm that solves the problem given the list of the number of pages for the books $L = \{b_1, \ldots, b_B\}$. Remember to analyse the time and space usage of your algorithm and argue for the correctness.

Solution:

The exam set continues on the next page

Solution 5.2 continued:

Question 6 (22%)

Question 6.1 (14%)

At the book club, you have a collection C of k different books. For some of the books you have many copies and for others only a few. There are m members of the book club. Each member j of the book club has made a list L_j of the books they would like to read. Each member should read 4 books before the next meeting. The book club has given you the responsibility of finding an assignment of books so that every member gets 4 books from their list home with them. For $1 \leq i \leq k$, let c_i denote how many copies of book i the book club has. You can assign book i to at most c_i members.

Give an algorithm that given $\{c_1, \ldots, c_k\}$ and $\{L_1, \ldots, L_m\}$ returns an assignment of books to members such that each member gets exactly 4 books. If this is not possible the algorithm should return "Not possible". Analyze the time and space usage of your algorithm in terms of k, m, and $C = \sum_{i=1}^{k} c_i$ and $L = \sum_{j=1}^{m} |L_j|$. Remember to argue that your algorithm is correct.

Solution:

Solution 6.1 continued:

Solution 6.1 continued:

The exam set continues on the next page

Question 6.2 (8%)

The books owned by the book club cover a large amount of different subjects. Two books are related if they have at least one subject in common. For example, a book about horse riding has the subject of horses in common with a book on horse breeding. The club wants to find the largest possible subset of their books such that no pair of books in the set has a subject in common. Let n be the number of books, let $T = \{t_1, \dots t_k\}$ be the set of subjects, and let $S_i \subseteq T$ be the subjects covered by book i for $1 \leq i \leq n$. Prove that this problem is NP-complete.

Solution:

The exam set continues on the next page

Solution 6.2 continued: