Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chintan Shah

Winter Semester 2011-12
Final Examination
25th February, 2012

EFFICIENT ALGORITHMS AND DATASTRUCTURES I

Last Name	First Name
....	....

Matrikel No.	Seat No.	Signature
....	....	....

General Information for the Examination

- Write your name and matrikel no. on all extra supplementaries (Blätter) provided.
- Please keep your identity card readily available.
- Do not use pencils. Do not write in red or green ink.
- You are not allowed to use any device other than your pens and a single sided handwritten A4 sized paper (with your name clearly written on top).
- Verify that you have received 13 printed sides (check page numbers).
- Attempt all questions. You have 120 minutes to answer the questions.

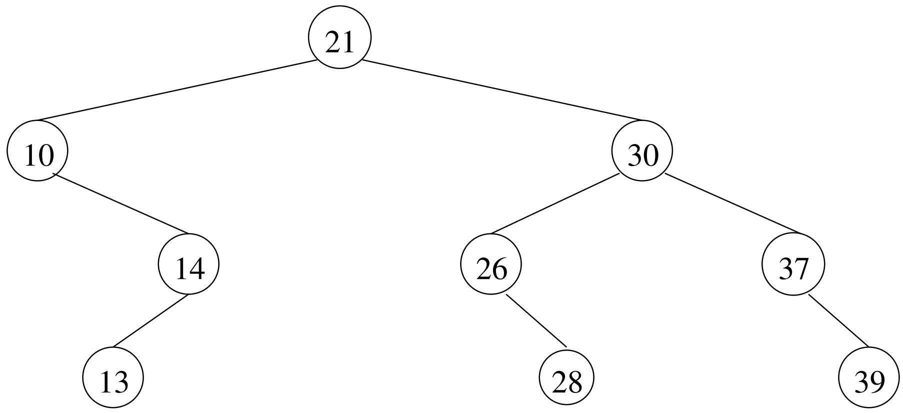
Left Examination Hall from ...... to ...... / from ...... to ......
Submitted Early at ......
Special Notes:

	A1	A2	A3	A4	A5	A6	A7	A8	Σ	Examiner
Maximum	2	2	3	5	6	6	8	8	40	
1st Correction										
2nd Correction										

Question 1 (2 Marks)

(a) Suppose that a maximum flow network contains a node, other than the sink node, with no outgoing edges. Can we delete this node without affecting the maximum flow value?

(b) Is the following tree an AVL tree?



Question 2 (2 Marks)

If we insert a node in a red-black tree and then immediately delete the same node,

Question 3 (3 Marks)

Show that in a disjoint-set implementation using both union by rank and path compression, any sequence of m MAKE-SET, FIND-SET and LINK operations takes only $O(m)$ time if all the LINK operations appear before any of the FIND-SET operations.

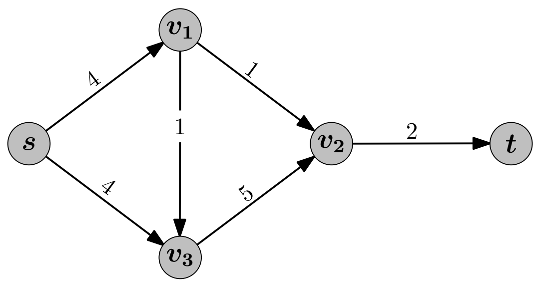
Question 4 (5 Marks)

An order-statistics tree is an augmented Binary Search Tree that supports the additional operations RANK(x), which returns the rank of x (i.e., the number of elements with keys less than or equal to x) and FINDBYRANK(k), which returns the kth smallest element of the tree.

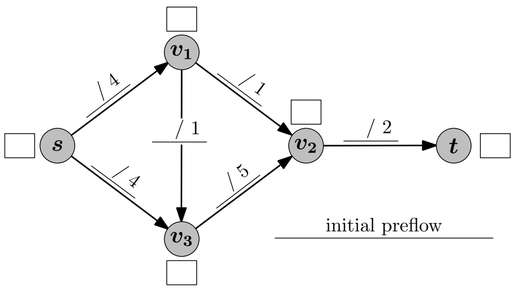
Let $A[1, \cdots, n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A. Show how to use an order-statistics tree to count the number of inversions in A in time $O(n \log n)$.

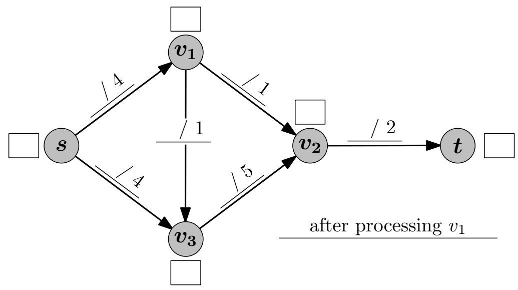
Question 5 (6 Marks)

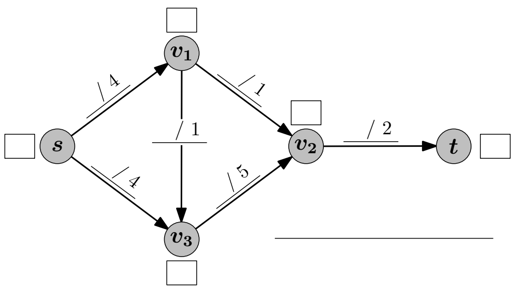
Simulate the preflow-push algorithm on the graph given below. First give the labels and flow values for each edge in the initial preflow. Then give the corresponding values each time the processing of a node is finished and a new active node is chosen.
Always choose an active node with highest label. In case of a tie, proceed in order of the nodes (i.e., first v_1, then v_2, then v_3). In case there are several admissible edges you are free to choose.

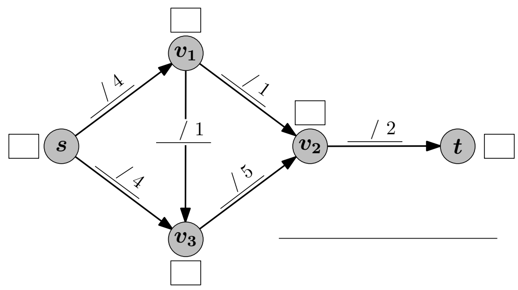


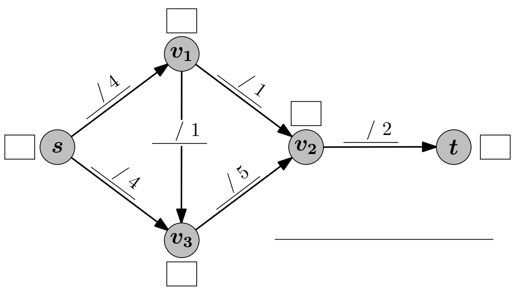
Use the graphs below. Note that the number of graphs below only indicates an upper bound on the number of steps required.

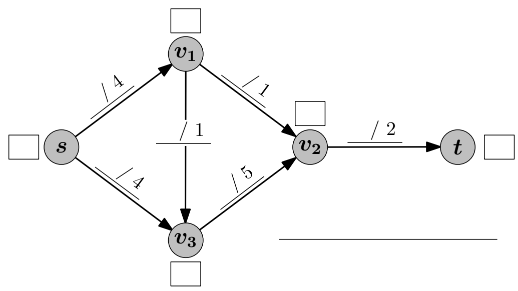




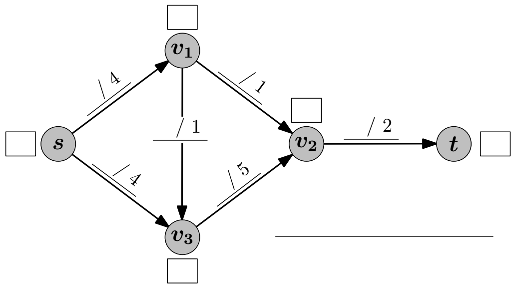


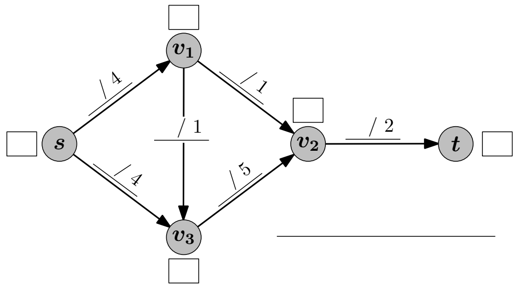


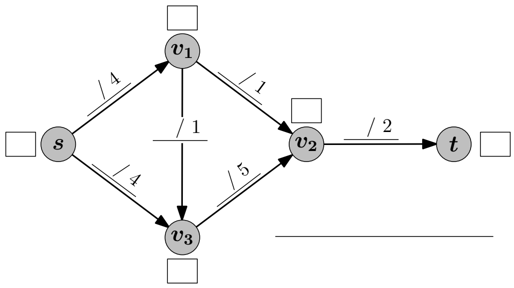


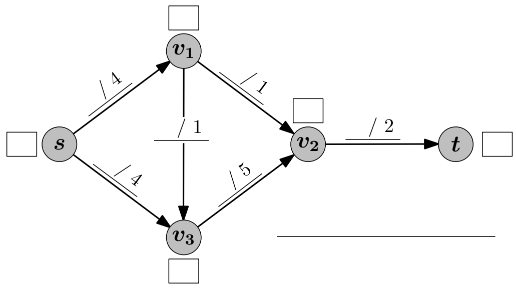


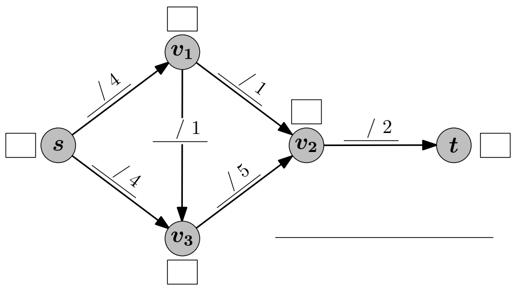
(Continue on next page . . .)

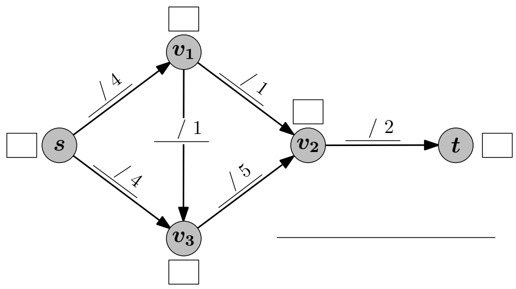

v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3


v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3


v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3


v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3


v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3


v1
/ 4
/ 1
/ 1
s
v2
/ 2
t
/ 4
/ 5
v3

ROUGH WORK

Question 6 (6 Marks)

Consider a 0-1 matrix A with n rows and m columns. We refer to a row or a column of the matrix A as a *line*. We say that a set of 1's in the matrix A is *independent* if no two of them appear in the same line. We also say that a set of lines in the matrix is a *cover* of A if they cover all the 1's in the matrix. Using the max-flow min-cut theorem, show that the maximum number of independent 1's equals the minimum number of lines in a cover.

Question 7 (8 Marks)

Suppose instead of using decimal or dual representation of numbers, we represent them in binary over the basis of Fibonacci numbers. That is, the bit-string $(X_k, X_{k-1}, \cdots, X_1)_F$ represents the number $n = \sum_{i=1}^{k} X_i \cdot F_i$, where F_i denotes the ith Fibonacci number ($F_1 = F_2 = 1$ and $F_i = F_{i-1}+F_{i-2}$ for $i \geq 3$). For example, $(31)_{10}$ can be represented by the bit string $(10100100)_F$ since $F_8+F_6+F_3 = (21)_{10}+(8)_{10}+(2)_{10} = (31)_{10}$, and also by the bit string $(10011011)_F$ since $F_8+F_5+F_4+F_2+F_1 = (21)_{10}+(5)_{10}+(3)_{10}+(1)_{10}+(1)_{10} = (31)_{10}$.

(a) Argue that we can represent any number $n \in \mathbb{N}_0$ like this. (2 marks)

(b) Describe an algorithm which performs increment and decrement operations in this representation in constant amortized time (starting from 0). Assume that flipping each bit requires one unit of work. (6 marks)
(*Hint:* Use a potential function that assigns potential depending on whether consecutive pairs of bits are similar. For example, if bit i and bit $i+1$ are equal/different, you may say they contribute one unit to the potential. Make sure that the potential of $(0)_F$ is zero units.)

Question 8 (8 Marks)

In the minimum flow problem, we wish to send the minimum amount of flow from the source to the sink, while satisfying the given lower and upper bounds on edge capacities, $\ell(e) \leq f(e) \leq u(e), \forall e \in E(G)$.

(a) Given a flow network G with source s and sink t, construct a flow network G' and run a given blackbox algorithm for solving the maximum flow problem on G', to find whether G admits a feasible flow (a feasible flow is a flow satisfying the capacity bounds for each edge). (4 marks)

(b) If G admits a feasible flow, show how to solve the minimum flow problem, again using a given blackbox algorithm for solving the maximum flow problem. (4 marks)

ROUGH WORK

ROUGH WORK

ROUGH WORK