Reserved Cells

Ex. 1	
Ex. 2	
Ex. 3	
Ex. 4	
Ex. 5	
Ex. 6	
Tot.	

Algorithms and Programming

29 January 2018

Part I: Theory

Register Number _____ Family Name _____ First Name _____

Course: ◯ 01$OGDLP$ 10 credit ◯ 02$OGDLM$ 12 credit

No books or notes are allowed. Solve exercises directly within the reserved space. Added sheets are accepted only when strictly necessary. Examination time: 50 **minutes.**

1. (2.0 **points**)
   Given the following sequence of pairs, where the relation i-j means that node i is adjacent to node j:

   2-5 1-3 3-4 5-1 4-0 6-3 5-6 1-9

   apply an on-line connectivity algorithm with quick find, showing at each step the contents of the array and the forest of trees at the final step. Node names are integers in the range from 0 to 10.

2. (2.5 **points**)

10 **credit course** (01$OGDLP$)

Given the following sequence of integers stored in an array:

1 10 72 41 71 0 8 55 91 14 32 19 13 73 7 3 31

perform the first 2 steps of quick-sort to sort the array in **ascending** order. At each step indicate the pivot element you selected. Steps must be improperly considered in breadth on the recursion tree, rather than in depth. Return as a result the two partitions of the original array and the two partitions of those found at the previous step.

12 **credit course** (02$OGDLM$)

Given the following sequence of integers stored in an array:

1 10 72 41 71 0 8 55 91 14 32 19 13 73 7 3 31

perform the first 2 steps of quick-sort to sort the array in **descending** order. At each step indicate the pivot element you selected. Steps must be improperly considered in breadth on the recursion tree, rather than in depth. Return as a result the two partitions of the original array and the two partitions of those found at the previous step.

3. (1.0 **point)**

10 **credit course (**$01OGDLP$**)**

Convert the following expression from in-fix to post-fix notation:

$$(A - B)/\{(C/D) + [(D/(E - F)]\}$$

12 **credit course (**$02OGDLM$**)**

Convert the following expression from in-fix to post-fix notation:

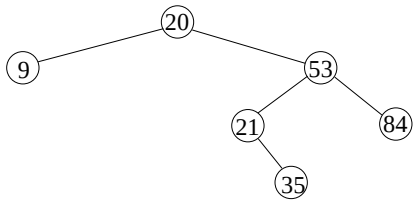
$$(A - B)/\{(C/D) + [(D/(E - F)) * G]\}$$

4. (2.0 **points**)
10 **credit course** (01$OGDLP$)
Consider a binary tree with 11 nodes. Its visits return the following sequences:

pre-order:	A	B	D	E	G	H	C	F	I	L	M
in-order:	D	B	G	E	H	A	F	L	I	M	C
post-order:	D	G	H	E	B	L	M	I	F	C	A

Draw the original binary tree.

12 **credit course** (02$OGDLM$)
Given the following BST



insert in the **root** the keys 16, 19 and 22; then delete the key 19. Redraw the tree at each relevant step.

5. (2.5 **points**)

10 **credit course** (01$OGDLP$)

Suppose to have an initially empty priority queue implemented with a **minimum** heap. Given the following sequence of integers and $\star$ characters:

$$11 \quad 23 \quad 5 \quad 7 \quad 13 \quad 25 \quad * \quad * \quad 2 \quad 31 \quad * \quad 19 \quad 17$$

where each integer corresponds to an insertion into the priority queue and each character $\star$ corresponds to an extraction of the maximum, show the priority queue after each operation and return the sequence of extracted values. Finally, change the priority of 17 into 41 and draw the corresponding priority queue.

12 **credit course** (02$OGDLM$)

Suppose to have an initially empty priority queue implemented with a **maximum** heap. Given the following sequence of integers and $\star$ characters:

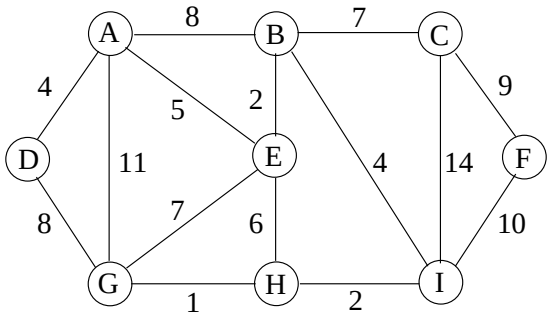
$$11 \quad 23 \quad 5 \quad 7 \quad 13 \quad 25 \quad * \quad * \quad 2 \quad 31 \quad * \quad 19 \quad 17$$

where each integer corresponds to an insertion into the priority queue and each character $\star$ corresponds to an extraction of the minimum, show the priority queue after each operation and return the sequence of extracted values. Finally, change the priority of 17 into 41 and draw the corresponding priority queue.

6. (2.0 **points**)

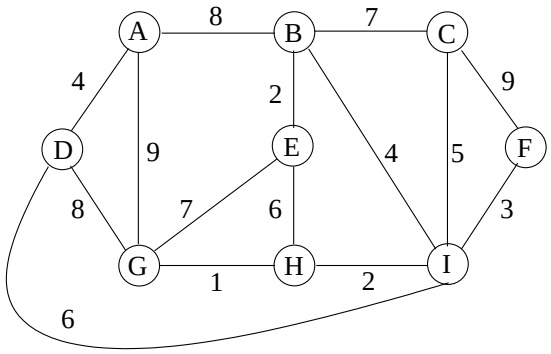
10 **credit course** (01$OGDLP$)

Given the following undirected and weighted graph, find a minimum spanning tree using Kruskal's algorithm, draw the tree and return the minimum weight as a result. Show relevant intermediate steps.



12 **credit course** (02$OGDLM$)

Given the following undirected and weighted graph, find a minimum spanning tree using Prim's algorithm starting from vertex D, draw the tree and return the minimum weight as a result. Show intermediate steps and all generated cuts.



Algorithms and Programming
29 January 2018
Part II: Program (12 point version)

At most one C manual is allowed. Examination time: 100 minutes. Final program due by 1.00 p.m. of Thursday the 1st of February; use the course portal page ("Elaborati" section) to upload it.

1 (2.0 points)
Write function

```
void subvet_write (int *v, int n);
```

which receives an array `v` storing `n` integer values, and it prints out (on standard output) all sub-arrays of maximum length storing *only* non-zero values.
For example let

1	0	2	3	-1	0	4	5	0	-2	3	4	8	0	0	0	3	-1	10	6

be the array v, and $n = 20$ its size. The function has to print sub-arrays $\{-2, 3, 4, 8\}$, and $\{3, -1, 10, 6\}$

2 (4.0 points)
A list stores integer values sorted in ascending order.
Write function

```
void list_complete (list_t *p);
```

which receives the reference `p` to the head of the list and it completes such a list, i.e., it adds one element to the list for each integer value not present in the original list and included between the minimum and maximum value stored into the same list.
For example, let us suppose that the list initially passed to the function stores values $\{4, 7, 10\}$. The list has to be modified by the function into the list $\{4, 5, 6, 7, 8, 9, 10\}$.
The candidate must also define the structure of the node of the list.

3 (6.0 points)
An activity a_i is characterized by an open time interval (s_i, f_i), where s_i is the starting time, f_i the finishing time, and $d_i = (f_i - s_i)$ is the duration of the time interval of a_i.
Given a set S of n activities $S = \{a_1, a_2, \ldots, a_n\}$ we want to write a program able to select the subsets of S such that no two activities overlap in time and the sum of their duration is maximum. In other words, we want to select the subset $\widehat{S}$ of S, such that:

- For each pair of activities i and j in such a subset there is no overlap: $\forall_{a_i, a_j \in \widehat{S}} \; (f_i \leq s_j \;\; OR \;\; f_j \leq s_i)$
- The sum of the length of the selected activities is maximum: $\sum_{a_i \in \widehat{S}} d_i$ is maximum.

Write function

```
void activity_selection (int **m, int *v, int n);
```

which receives a matrix `m` storing on `n` rows the set of activities, and returns in the array `v` (with `n` elements), v[i] = 1 if the activity i has been selected, and v[i] = 0 otherwise. Notice that `m` has `n` rows and `2` columns, representing s_i and f_i, respectively.
For example if `n` = 5, and `m` = $\{(1, 4), (3, 5), (4, 6), (5, 9), (8, 9)\}$ the function must select activities $(1, 4)$ and $(5, 9)$ which do not overlap and have a total duration equal to $(4 - 1) + (9 - 5) = 7$. Then the function has to set the array `v` as `v` = $\{1, 0, 0, 1, 0\}$.

Algorithms and Programming
29 January 2018
Part II: Program (18 point version)

At most one C manual is allowed. Examination time: 100 minutes. Final program due by 1.00 p.m. of Thursday the 1st of February; use the course portal page ("Elaborati" section) to upload it.

A temporary work agency enrolled N employees looking for a new employer and N firms looking for a new employee. The target of the temp agency is to allocate a new employee to each firm, and to maximize the overall employee-and-firm satisfaction. To do that it wants to design the following algorithm.
The temp agency represents all employees and firms using their alphanumeric names (or identifiers) of at most 20 characters. Then, each employee ranks all N firms in order of preference, and each firm ranks all employees in order of preference.
For example, let us suppose that $N = 4$, the employee names are $\{e1, e2, e3, e4\}$ and the firm names are $\{f1, f2, f3, f4\}$. Then the two matrices:

e1	f2	f4	f1	f3
e2	f3	f1	f4	f2
e3	f2	f3	f1	f4
e4	f4	f1	f3	f2

f1	e2	e1	e4	e3
f2	e4	e3	e1	e2
f3	e1	e4	e3	e2
f4	e2	e1	e4	e3

state that, $e1$ ranked firms in preference order as $f2$, $f4$, $f1$, $f3$. At the same time, firm $f4$, for example, ranked employees in preference order as $e2$, $e1$, $e4$, $e3$.

In this scenario given a possible match for all employees on all firms, any pair employee-firm (e, f) out of the possible N^2 pairs is said to be a source of *instability* for the given match *if and only if* both the following conditions are true:

- The employee e prefers the firm f to the one it has been assigned to.
- The firm f prefers the employee e to the one it has been assigned to.

For example, given the match $\{(e1, f1), (e2, f3), (e3, f2), (e4, f4)\}$ the pair $(e1, f4)$ is a source of instability, because the employee $e1$ prefers firm $f4$ to firm $f1$ (the one assigned to $e1$ by the match), and firm $f4$ prefers employee $e1$ to $e4$ (the one assigned to $f4$ by the match)
On the contrary, the match $\{(e1, f4), (e2, f3), (e3, f2), (e4, f1)\}$ has no instability pairs.

Write a program that:

- Receives a file name on the command line. This file stores the value of N on the first row, and the two previously described matrices on the following $(2 \cdot N)$ rows. An example of such a file is represented in the following picture.

```
4
e1 f2 f4 f1 f3
e2 f3 f1 f4 f2
e3 f2 f3 f1 f4
e4 f4 f1 f3 f2
f1 e2 e1 e4 e3
f2 e4 e3 e1 e2
f3 e1 e4 e3 e2
f4 e2 e1 e4 e3
```

- Finds a possible match between employees and firms such that no employee-firm pair (e, f) is a source of instability. If such a match does exist print on standard output the employee-to-firm correspondence.