Reserved Cells

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

Algorithms and Programming

13 February 2018

Part I: Theory

Register Number _____ Family Name _____ First Name _____

Course: ○ 01*OGDLP* 10 credit ○ 02*OGDLM* 12 credit

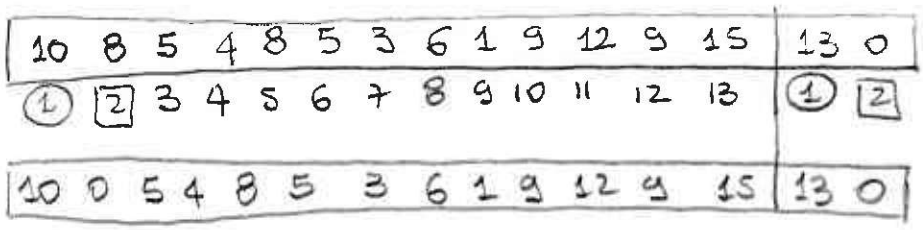
No books or notes are allowed. Solve exercises directly within the reserved space. Added sheets are accepted only when strictly necessary. Examination time: 50 minutes.

1. (2.0 **points**)
   Sort in ascending order with shell sort the following array of integers:

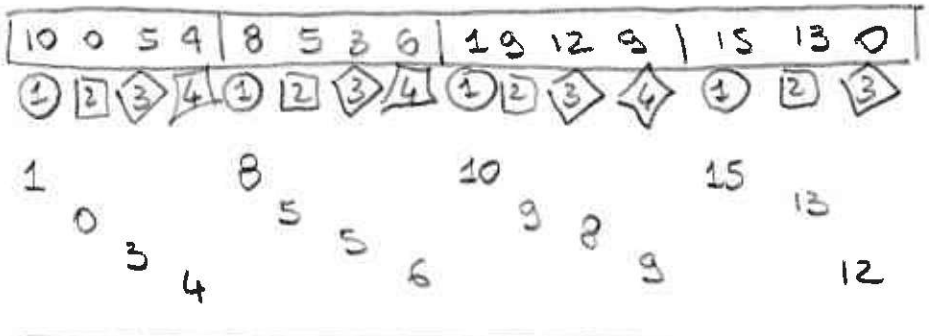
   10 8 5 4 8 5 3 6 1 9 12 9 15 13 0

   Use the Knuth's sequence $h = 3 \cdot h + 1$ (1, 4, 13, etc.). Report all main steps to obtain the final order.

h = 13



h = 4



h = 1

0 1
0 1 3 4 8
0 1 3 4 5 8
0 1 3 4 5 5 8
0 1 3 4 5 5 6 8 10
0 1 3 4 5 5 6 8 9 10
0 1 3 4 5 5 6 8 8 9 10
0 1 3 4 5 5 6 8 8 9 9 10 15
0 1 3 4 5 5 6 8 8 9 9 10 13 15
0 1 3 4 5 5 6 8 8 9 9 10 12 13 15

2. **(1.0 points)**
Sort in ascending order with counting sort the following array of integers:

1 4 5 3 2 7 8 2 1 9 0 2 6 9 3

Show the content of the arrays A, B and C, and all relevant intermediate steps on the array C.

A

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	5	3	2	7	8	2	1	9	0	2	6	9	3

	0	1	2	3	4	5	6	7	8	9
C_1	0	0	0	0	0	0	0	0	0	0
C_2	1	2	3	2	1	1	1	1	1	2
C_3	1	3	6	8	9	10	11	12	13	15
	0	2	5	7	8	9	10	11	12	14
		1	4	6						13
			3							

B

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	1	2	2	2	3	3	4	5	6	7	8	9	9

$B[C[A[j]]-1] = A[j]$

Copy B into A

3. (2.0 **points**)

10 credit course (01*OGDLP*)

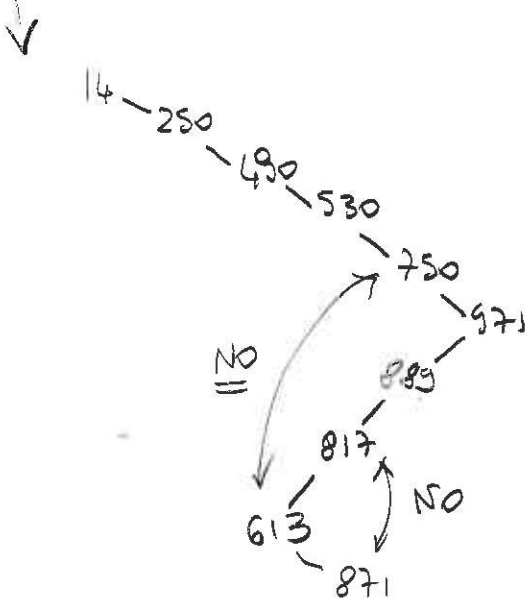
A correct BST contain integer keys in the range 1-1000. The user searches for key 871. Among these sequences, which are the ones that cannot occur? Why?

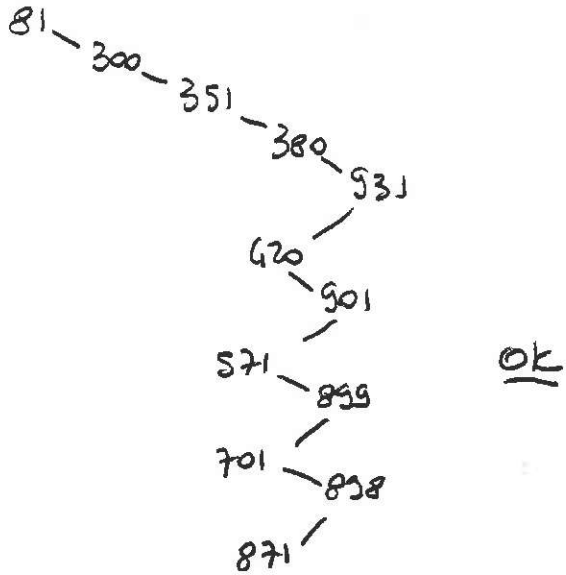
14	250	490	530	750	971	889	817	613	871		
81	300	351	380	931	420	901	571	899	701	898	871
998	100	900	310	390	400	600	700	800	899	871	
500	991	687	792	988	810	757	880	885	871		

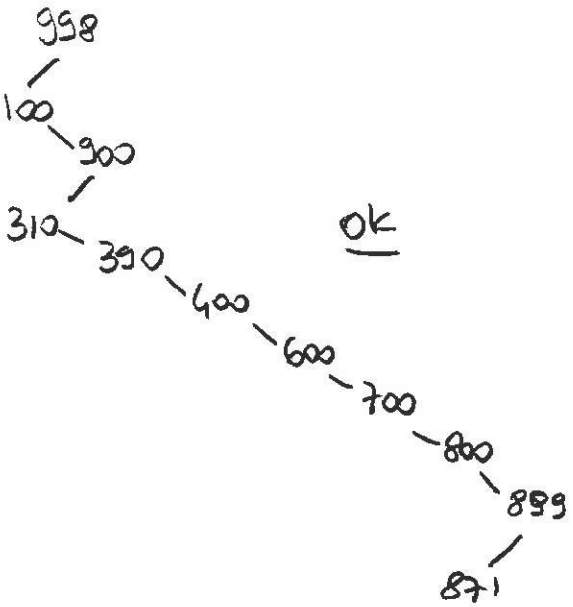
12 credit course (02*OGDLM*)

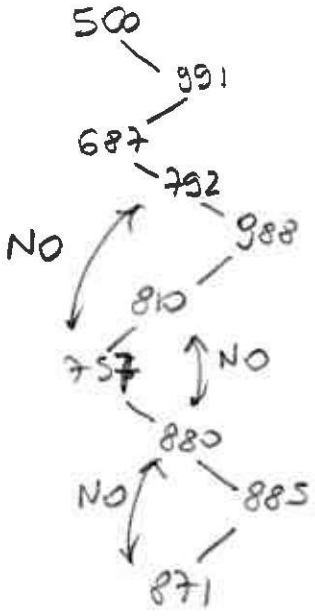
Using a greedy algorithm find an optimal Huffman code for the following symbols with the specified frequencies:

A:12 *B*:32 *C*:10 *D*:8 *E*:9 *F*:17 *G*:13 *H*:4 *I*:11 *J*:8



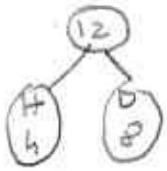




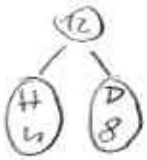


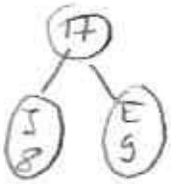
H D J E C I A G F B
4 8 8 9 10 11 12 13 17 32

J E C I A G F B
8 9 10 11 12 13 17 32

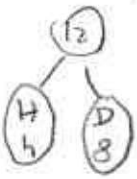


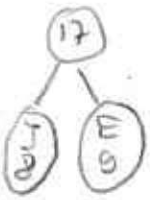
C I A G F B
10 11 12 13 17 32

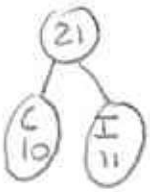




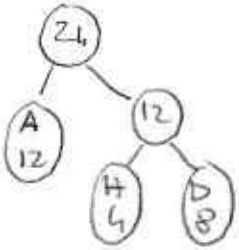
A G F B
12 13 17 32



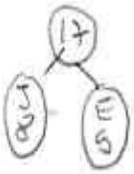


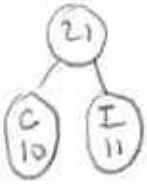


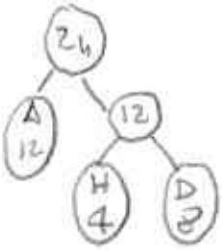
G F B
13 17 32

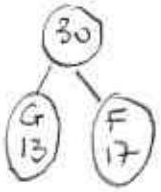


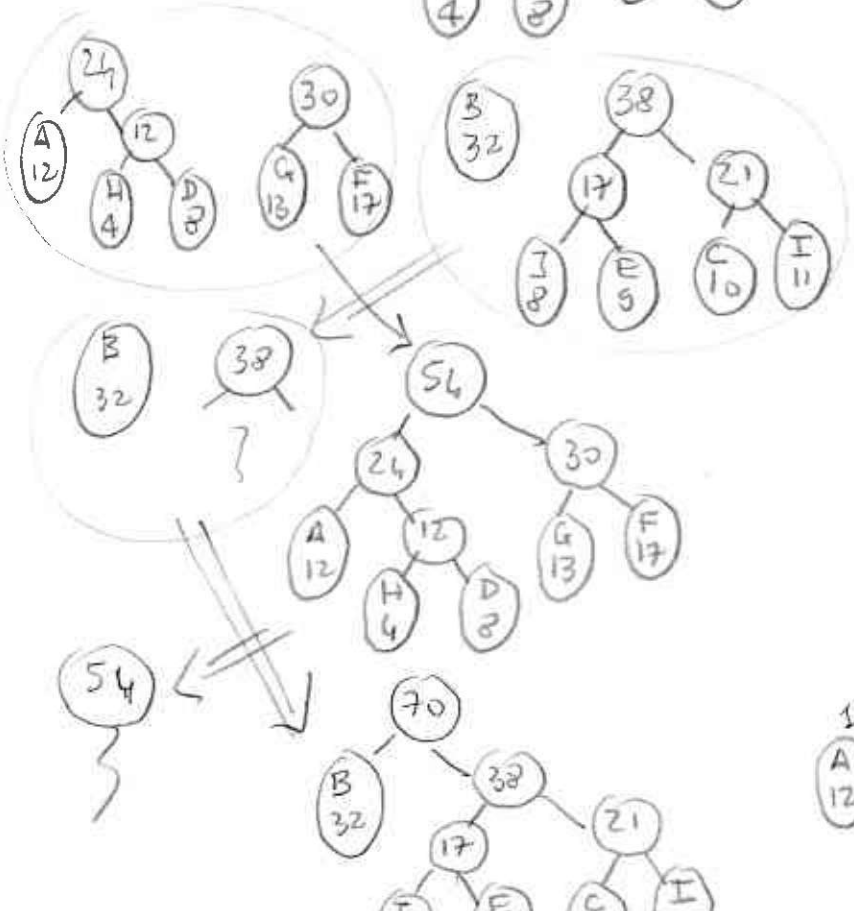
B
32



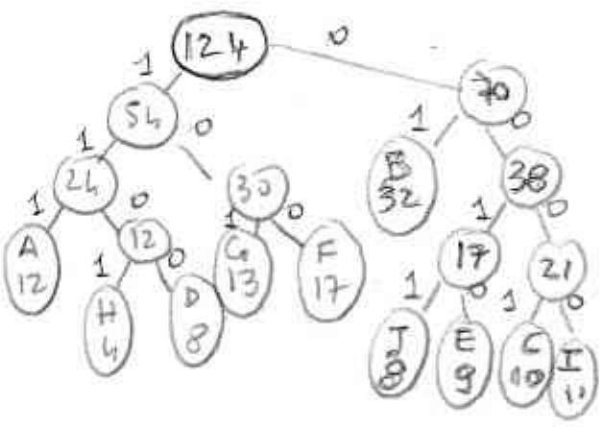








A 1 1 1
B 0 1
C 0 0 0 1
D 1 1 0 0
E 0 0 1 0
F 1 0 0
G 1 0 1
H 1 1 0 1
I 0 0 0 0
J 0 0 1 1



4. **(2.0 points)**
Given the sequence of keys

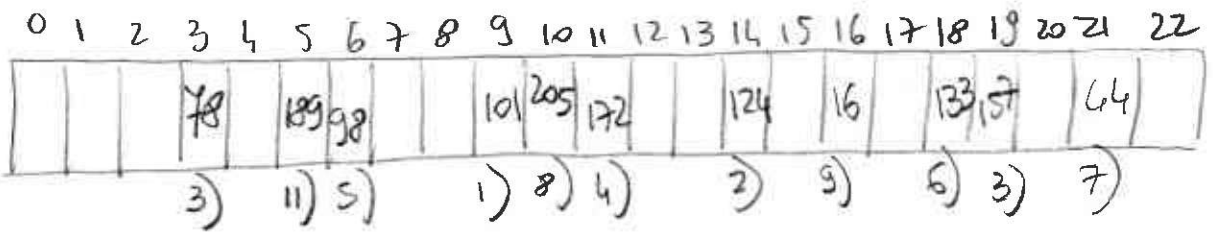
101 124 157 172 98 133 44 205 16 78 189

draw the final configuration of an initially empty hash table of size 23 where insertion of the previous sequence occurs. Assume open addressing with double hashing and hash functions $h_1(k) = k\%23$, $h_2(k) = 1 + k\%97$. Show relevant intermediate steps.

23x	1	2	3	4	5	6	7	8	9	10	...	14
=	23	46	69	92	115	138	161	184	207	230		322

$h(k) = [h_1(k) + i \cdot h_2(k)] \% 23 = [(k\%23) + i(1 + k\%97)] \% 23$

		$h_1 = k\%23$	$h_2 = 1+k\%97$	h
1)	101	9		
2)	124	9 ✱	1+27 = 28	(9+28) % 23 = 37 % 23 = 14
3)	157	19		
4)	172	11		
5)	98	6		
6)	133	18		
7)	44	21		
8)	205	21 ✱	1+11 = 12	(21+12) % 23 = 33 % 23 = 10
9)	16	16		
10)	78	9 ✱	1+78 = 79	(9+79) % 23 = 88 % 23 = 19 ✱ (9+2·79) % 23 = 167 % 23 = 6 ✱ (9+3·79) % 23 = 246 % 23 = 16 ✱
11)	189	5		(9+4·79) % 23 = 325 % 23 = 3

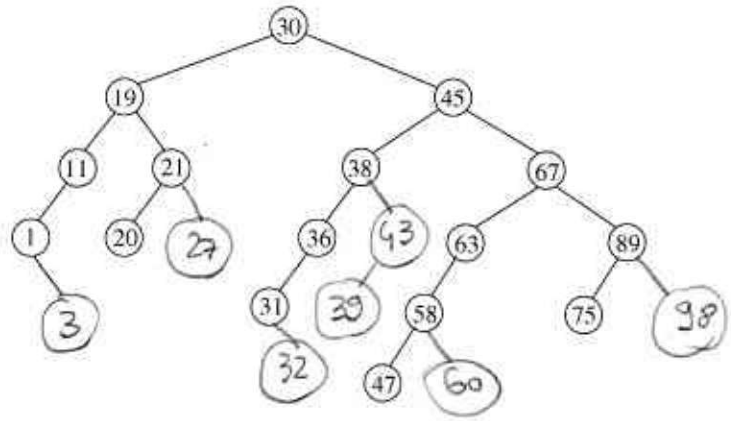


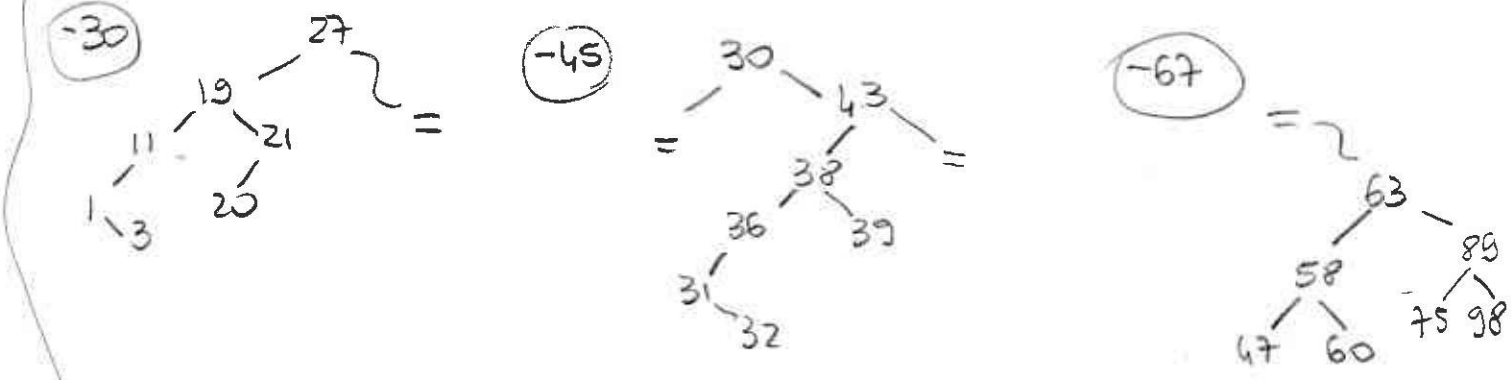
5. (1.0 **point**)
10 **credit course** (01*OGDLP*)
Given the following BST perform the following operations:

$$+3 \quad +27 \quad +98 \quad +43 \quad +39 \quad +60 \quad +32 \quad -30 \quad -45 \quad -67$$

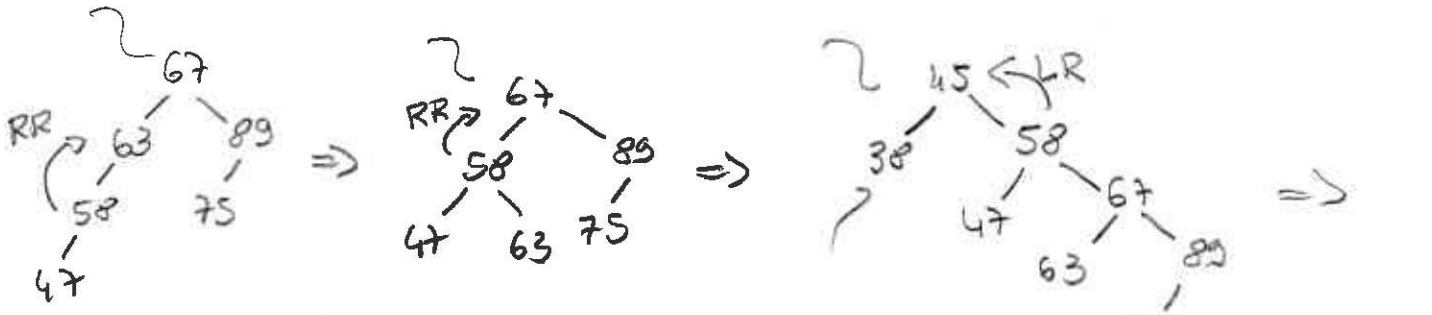
where each + symbol indicates and insertion in the leaves, and each − symbol represents an extraction.

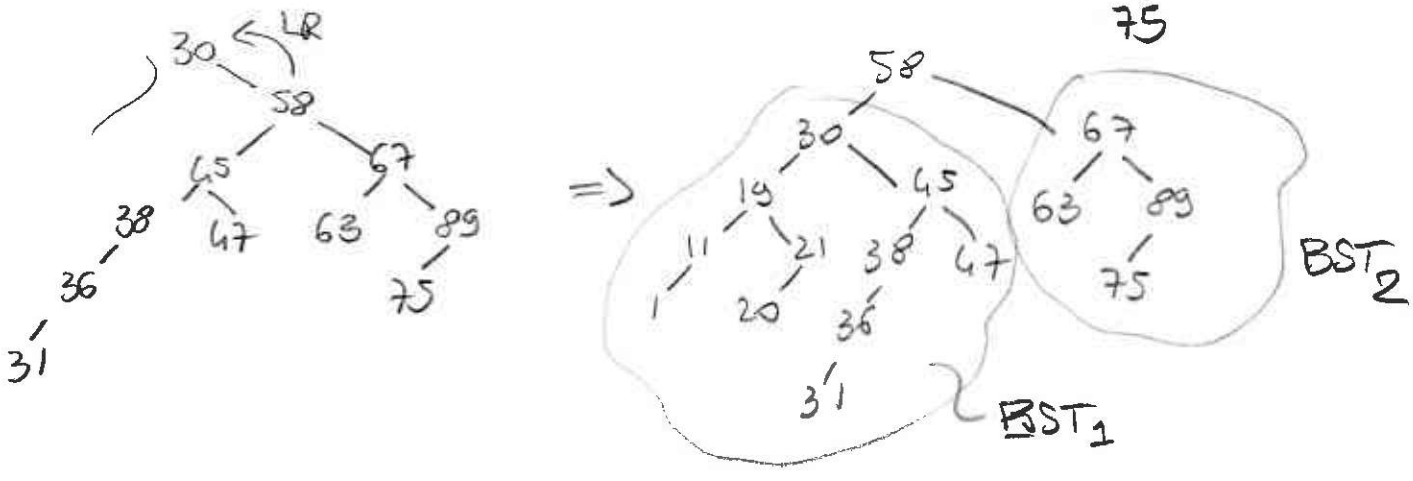
12 **credit course** (02*OGDLM*)
Partition the following BST around key 58:



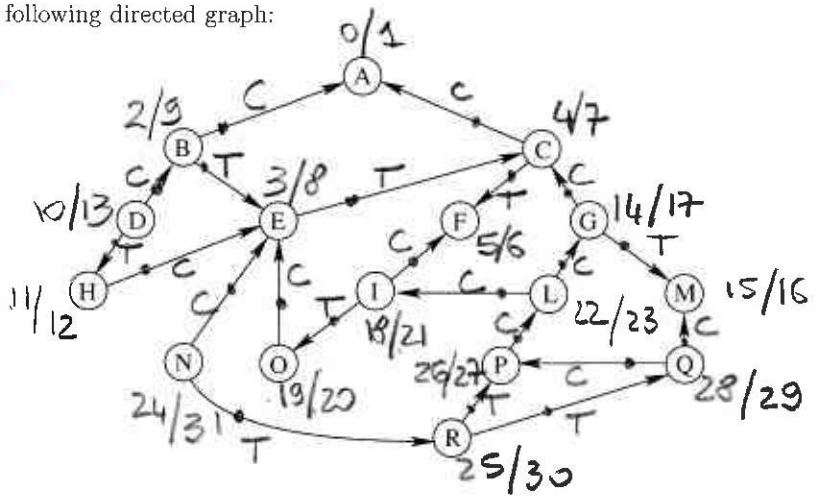


NONONO





6. **(1.0 + 2.0 + 1.0 points)**
Suppose to have the following directed graph:



- Represent it as an adjacency matrix.
- Visit it in depth-first starting at node A. Label nodes with discovery and end-processing times in the format $time_1/time_2$.
- Redraw it labeling each edge as T (tree), B (back), F (forward), C (cross).

Whenever necessary consider nodes in alphabetical order.