LULEÅ TEKNISKA UNIVERSITET

KURSKOD / COURSE CODE	PROV / TEST CODE
D0041D	0001

KURSBENÄMNING / COURSE NAME

Datastrukturer och algoritmer

PROVBENÄMNING / TEST NAME

Tentamen

TENTAMENSDATUM / EXAMINATION DATE

2021-06-03

TENTAMENSORT/CITY (för distansstudenter / for off campus students only)

ÅÅÅÅMMDD-XXXX
YYYYMMDD-XXXX

PERSONNUMMER / PERSONAL NUMBER

20000302-5095

NAMN (TEXTA) / FULL NAME

Morgan Nyman

NAMNTECKNING / YOUR SIGNATURE

PROGRAM	INSKR.ÅR/YEAR	ANTAL SIDOR / NO. OF PAGES

Skanningsblad/Scanning Sheet

Behandlat uppgift nr (sätt x) / Mark the questions you answered with an X			Lärarens anteckningar / Teacher's notes
1	X	12	
2	X	2	
3	X	15	
4	X	3,0	
5	X	7,0	
6	—	0	
7	—	0	
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
	Poängsumma / Points	39	Betyg / Grade: U

23294

Tentamensomslag skall alltid inlämnas även om ingen uppgift behandlats
Examination cover should always be submitted even if no questions are answered

Utskriven 2021-06-07 kl. 12:45:50

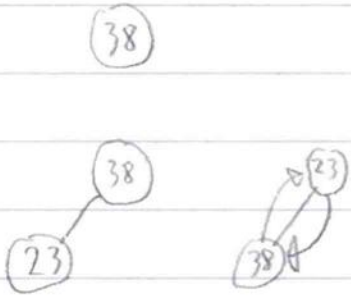
Sida 1
uppgift 1

DD041D

Morgan Nyman
000302-5095
2021-06-03

a) min-heap

13	18	23	38	28
0	1	2	3	4

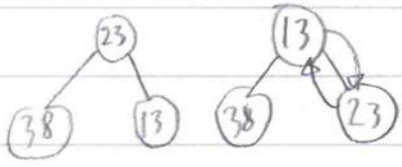


~~38~~, ~~23~~, ~~13~~, ~~38~~, ~~18~~

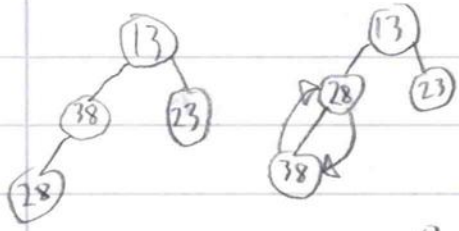
I insert new node at the last index

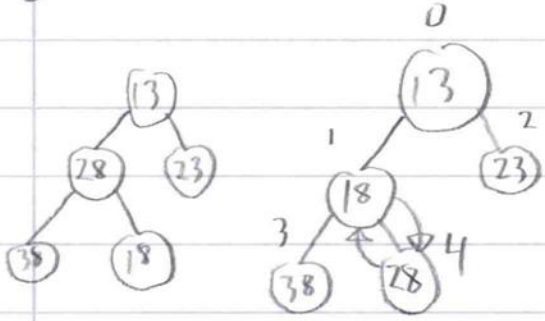
II swap the smallest child node with it's parents all the way up the tree (heapify)

III repeat



(Note: all parent nodes should be less than or equal to their child nodes)



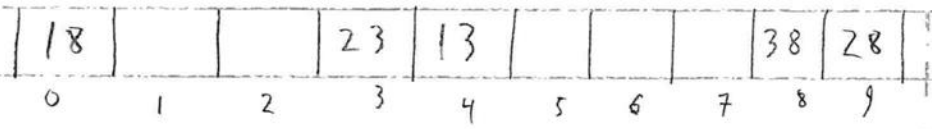


Sida 2
uppgift 1

DD041D

Morgan Nyman
0003 02-5095
2021-06-03

b)



38, 23, 13, 28, 18

I attempt to put the value at the index given by h(x,i) if index is occupied, increment i and run h(x,i) again (linear probing)

$h(x, i) = (x \bmod 10 + i) \bmod 10$

$h(38,0) = 8$ insert

$h(23,0) = 3$ insert

$h(13,0) = 3$ occupied

$h(13,1) = (13 \bmod 10 + 1) \bmod 10 = 4$ insert

$h(28,0) = 8$ occupied

$h(28,1) = (28 \bmod 10 + 1) \bmod 10 = 9$ insert

$h(18,0) = 8$ occupied

$h(18,1) = (18 \bmod 10 + 1) \bmod 10 = 9$ occupied

$h(18,2) = (18 \bmod 10 + 2) \bmod 10 = 0$ insert

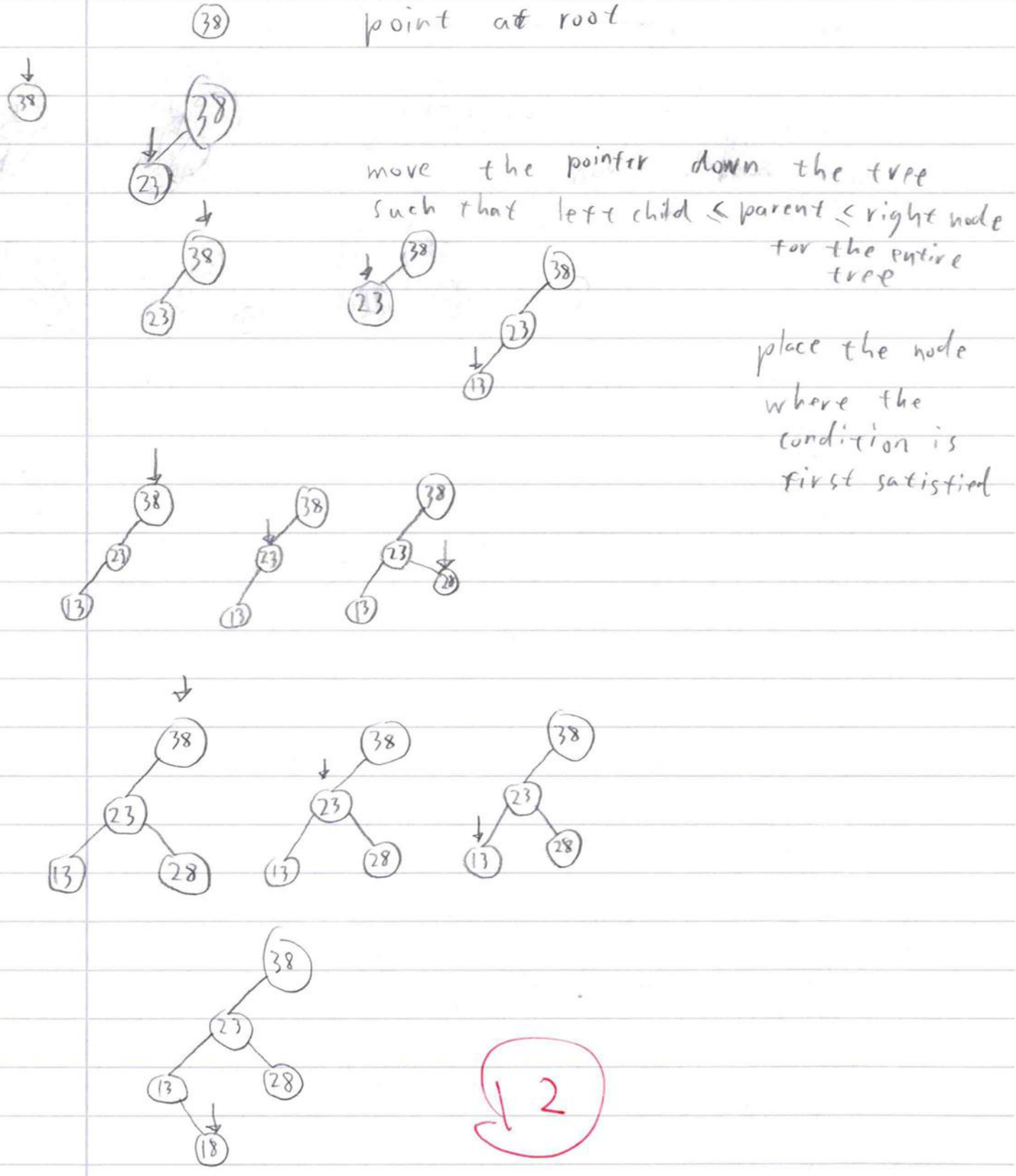
-- end --

Sida 3
uppgift 1

DD041D

Morgan Nyman
000302-5095
2021-06-03

c) binary search tree 38, 23, 13, 28, 18



12

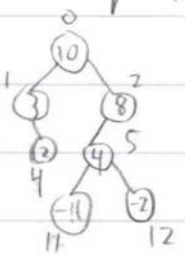
Uppgift 2
sida 4

DD041D

Morgan Nyman
000302-5095
2021-06-03

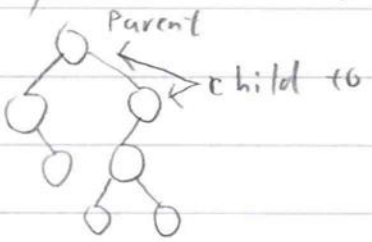
a) (2)

max heap example



not a heap →

binary tree example



-2p traverse the tree T. check that parent of the binary tree T is greater than or equal to it's children (or in array form: index $n \in T$, $n \geq 2n+1$ and $n \geq 2n+2$).
If at least one node return false on the check. T is not a Max heap, if the search ends with all conditions met, you have a max-heap.

b) def isBinSearchTree(T):

-4p X

```
    for n in range(len(T)): # as an array
        if 2*n+1<=n<=2*n+2:
            continue
        else:
            return false
    return true
```

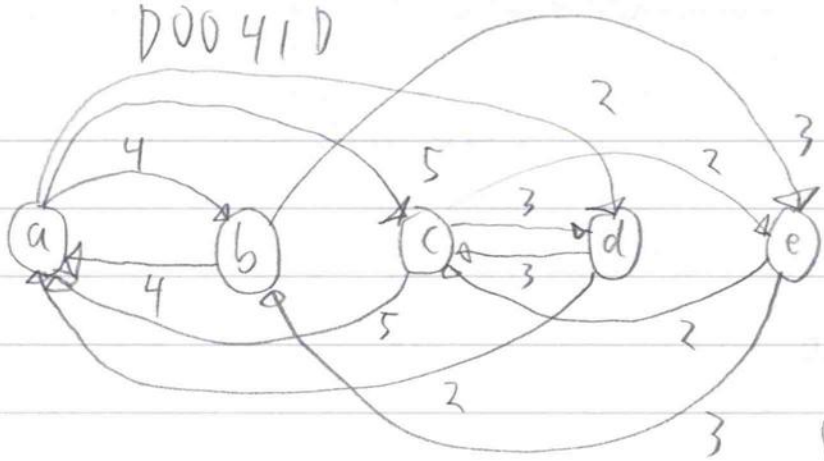
isBinSearchTree(T)

Sida 5
uppgift 3

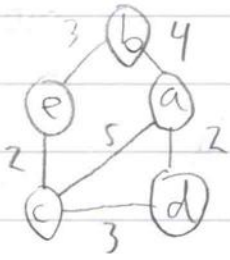
DD0410

Morgan Nyman
000302 5095
2021-06-03

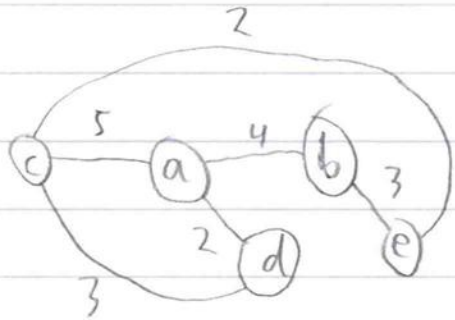
a)



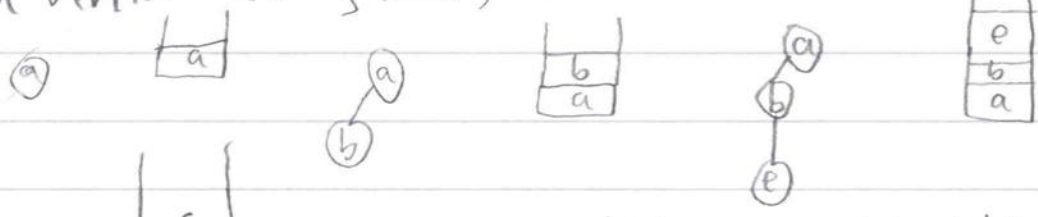
the graph



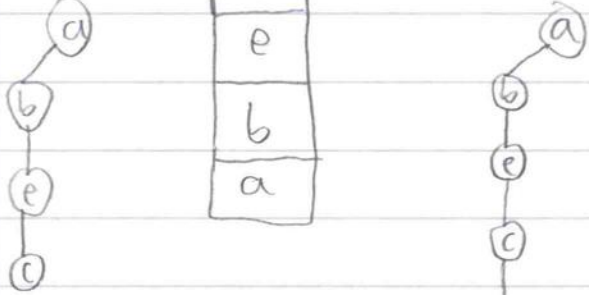
15

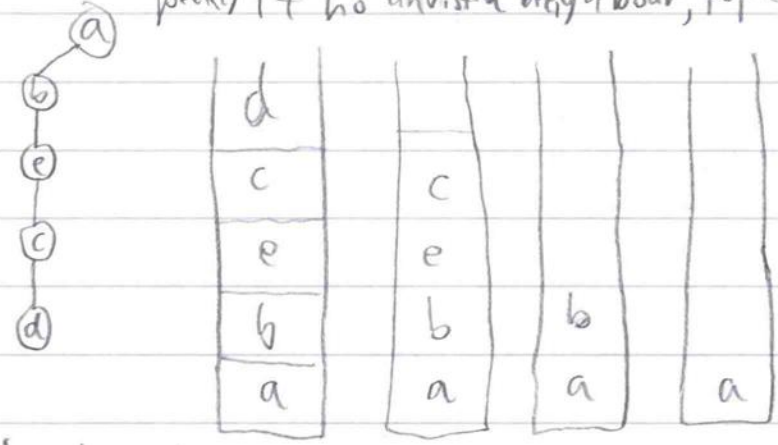


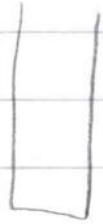
b) add vertice to stack, move to oneof it's neighbour



peek() if no unvisited neighbour, pop()







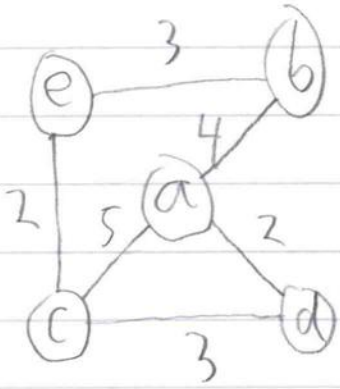
when all verticies has been popped from the stack the DFS returns the path.

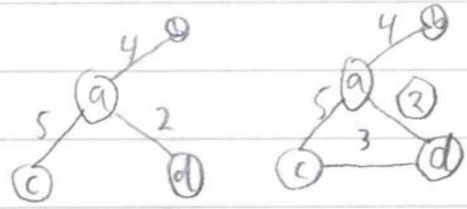
Sida 6
uppgift 3

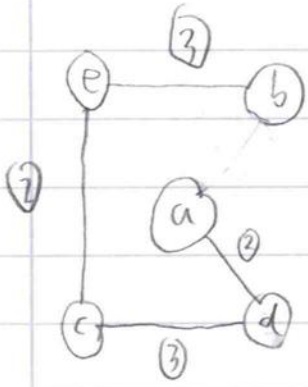
DOO41D

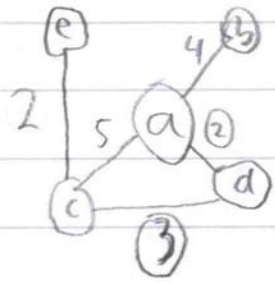
Morgan Nyman
000302-5095
2021-06-03

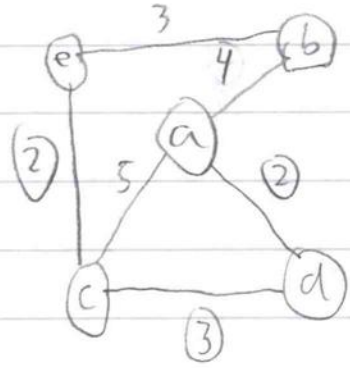
c) i. prims algorithm, start at (a). I add the shortest edge not in to a node that is in the tree. I reveal the new edge cases.











Prim order

order: a-d , d-c , c-e , e-b
0 1 2 3

ii. kruskal's algorithm add the lowest weight edge to the spanning tree. it both of it's neighbour isn't already in the MST

kruskal's order: a-d , c-e , b-e , c-d
0 1 2 3

Sida 7
Uppgift 4

Morgan Nyman
000302-5095
2021-06-03

3.0

a) False

b) ~~$\Theta(E)$~~ for $G = (V, E)$ — 3p

c) ~~True~~ — 3p

d) ~~$\Theta(n \log n)$~~ — 5p

e) ~~$O(2^n)$~~ in the worst case — 5p

f) ~~True~~ — 3p

g) ~~False~~ and ~~$\Theta(k)$~~ — 5p

h) ~~False~~ — 3p

Sida 8
uppgift 5

D0041D

Morgan Nyman
000302-5095
2021-06-03

1) c

7.0

2) b

3) ~~c~~ -1p

4) ~~c~~ -1p

5) c