LULEÅ TEKNISKA UNIVERSITET

KURSKOD / COURSE CODE

D	0	0	4	1	D

PROV / TEST CODE

0	0	0	1

KURSBENÄMNING / COURSE NAME

Datastrukturer och algoritmer

PROVBENÄMNING / TEST NAME

Tentamen

TENTAMENSDATUM / EXAMINATION DATE

2020-08-28

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	13.0	
2	X	0	
3	X	4	
4	X	9	
5	X	10	
6	X	0	
7	X	3	
8	—	0	
9	—	0	
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 2020-08-31 kl. 12:18:50

1. a) 24, 13, 17, 34, 15, 27, 15, 29, 22, 26 divide

24, 13, 17, 34, 15 divide

24, 13, 17 divide

24, 13 divide

24 | 13 sort and merge

13 24 | 17 sort and merge

13 17 24 | 34, 15 divide

34 | 15 sort and merge

13 17 24 | 15 34 sort and merge

27, 15, 29, 22, 26 divide

27, 15, 29

27, 15 divide

27 | 15 sort and merge

15 27 | 29

15 27 29 | 22, 26 divide

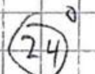
22 | 26 sort and merge

15 27 29 | 22 26

13 15 17 24 34 | 15 22 26 27 29 sort and merge

13 15 17 22 24 26 27 29 34 sorted!

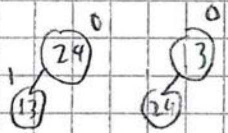
24, 13, 17, 34, 15 insert into min heap 1. b)



0
24

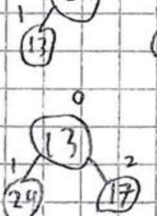
13, 17, 34, 15

heapify



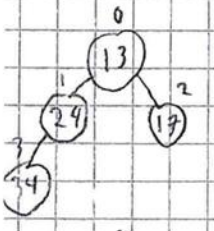
0	1
13	24

17, 34, 15



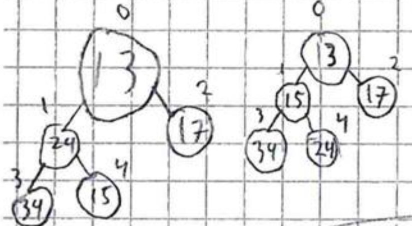
0	1	2
13	24	17

34, 15

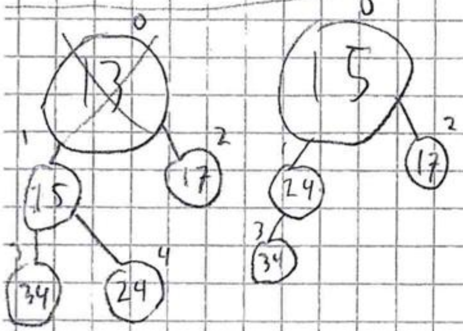


0	1	2	3
13	24	17	34

15



0	1	2	3	4
13	15	17	34	24



0	1	2	3	4
13				

perform min-delete
until heap is empty

200828 Morgan Nyman

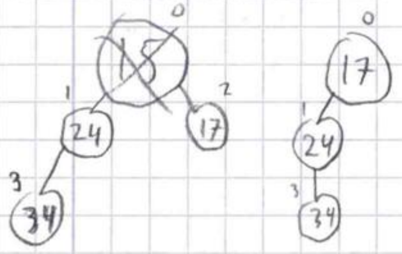
000302-5095

D0041D

Page 3

13.0

1. b)



0	1	2	3	4
13	15			

perform min-delete until heap is empty

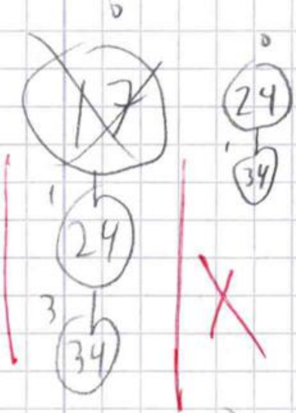
Not heap

heap on 24, 13, 17, 34, 15 ?

(OK)

~~24~~

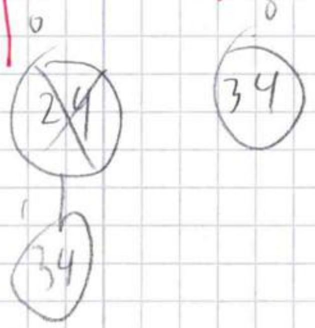
R



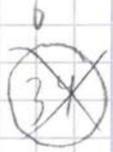
0	1	2	3	4
13	15	17		

-.2p

all the elements should be in the array all the time.



0	1	2	3	4
13	15	17	24	



nullptr

0	1	2	3	4
13	15	17	24	34

sorted!

1. c) merge sort: $\Theta(n \log n)$ in the worst case
(log n to divide, n to sort log n times, log n to merge)

heap sort: $\Theta(n \log n)$ in the worst case

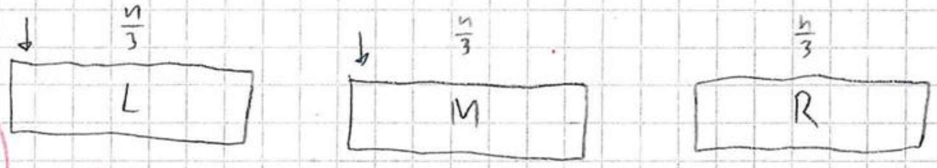
(n to construct heap, log n to heapify, 1 to return delete, log n to heapify)

200828 Morgan Nyman
000302-5095

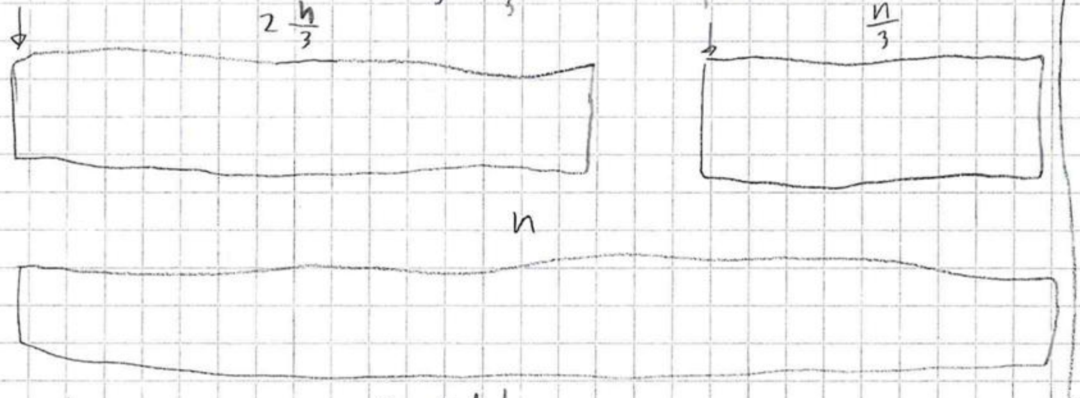
Page 4

D 0041 D

2. a)



using pointers, this can be achived in $O(n)$ comparisons in the worst case, start by comparing two sub-sequences than do the same for the new $\frac{2n}{3}$ sequence and compare with the remaining $\frac{n}{3}$ sub-sequence



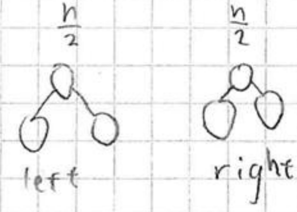
sorted!

answer:
a faster way is to make 3 comparisons in the initial state, L<M and R<M and L<R $\frac{n}{3}$ times, whic is $3\frac{n}{3} = n$
n comparisons

comparisons: $\left(\frac{n}{3}+\frac{n}{3}\right)+\left(\frac{2n}{3}+\frac{n}{3}\right) = \frac{5n}{3} = O(n)$

X — 4p

2. b)



compare the roots of the subtree in $O(1)$

set the root as root to the whole tree of size n

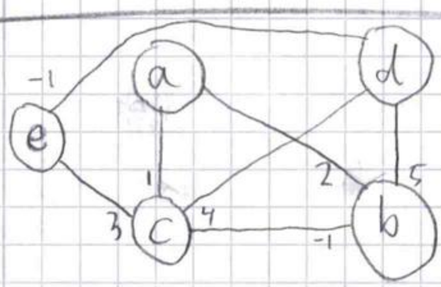
heapify the left + right subtrees in $\log n + \log n$ comparisons to make sure that the heap property is maintained. this the reason behind the $2\log n$ comparisons.

-4p. X

200828 Morgan Nyman

000302-5095 D0041D Page 5

3. a)



4

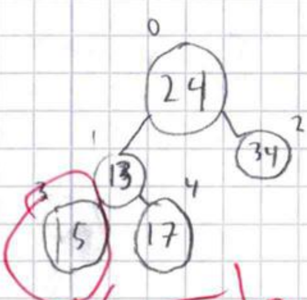
b) -4p

3. c) Dijkstra's algorithm does not operate on graphs with negative weighted edges, but Bellman Ford's algorithm does just that, therefore, we use Bellman Ford's algorithm to find the single-source shortest path.

X -2p.

4. 1) 24, 13, 17, 34, 15 a binary tree has a sorted list if traversed in order

9



X -1p

2) 24, 13, 14, 34, 15

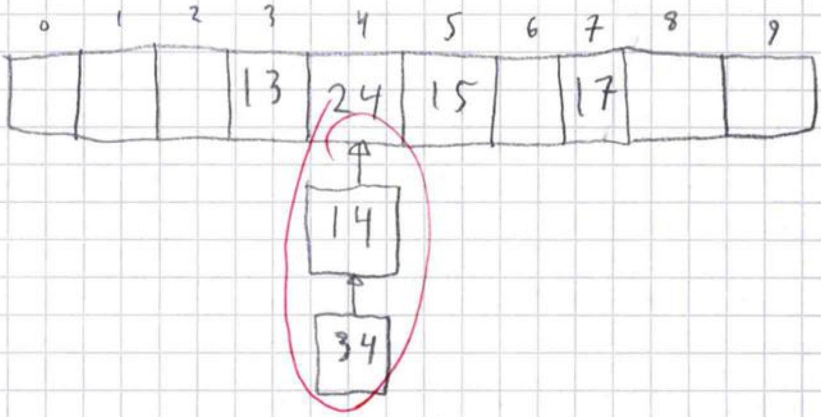
h(24) = 24 mod 10 = 4

h(13) = 13 mod 10 = 3

h(17) = 14 mod 10 = 4

h(34) = 34 mod 10 = 4

h(15) = 15 mod 10 = 5



34 → 14 → 24

-1p

a) to insert a single element into a binary search tree. 4.3)

you have to make sure that going down from root. if righ child < element: swap with right child
else if left child > element; swap with left child
if it's the end or both right > element > left
then it's finished, this process takes O(log n) comparisons in the worst case. answer: the statement is true.

because when going down the tree with every comparison done, you have half the nodes remaining as an option. dividing log n times leaves one reamai

−3p

b) using a prime number for the range is malpractice because a prime number has no factors. That makes multipling on collisions ineffecient. so no you must not use a prime number for the range, answer: False

c) True the left-most leaf of a binary search tree is the smallest element.

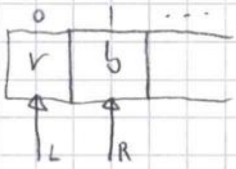
−3p

200828 Morgan Nyman
000302-5095 D0041D Page 7

10

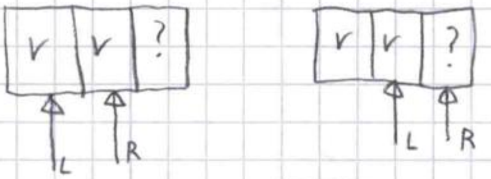
5. r = red (left) and $S = [r_1, r_2, r_3, \dots b_{n-2}, b_{n-1}, b_n]$
b = blue (right)

place two pointers at the start of the array side by side:



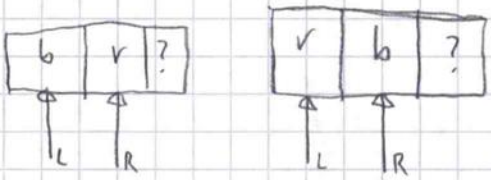
$\uparrow_L$ = left pointer
$\uparrow_R$ = right pointer

- if both pointers sees a red element then move both pointers along:



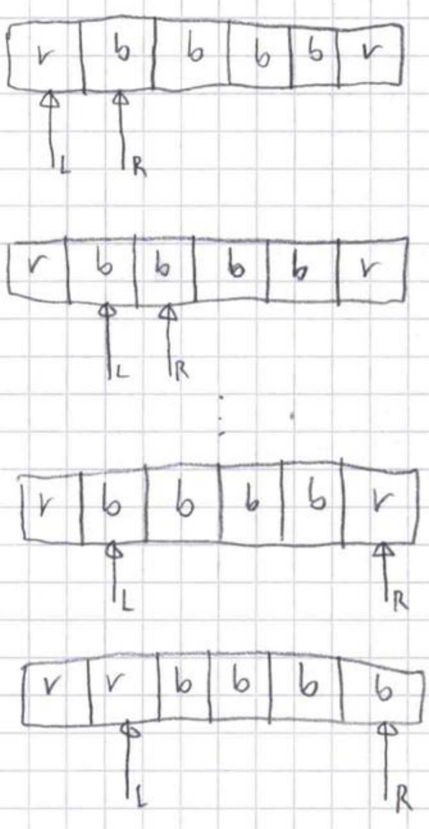
case: rr

- if the left pointer sees a blue element, keep moving the right pointer until the next red and swap the two colors, finally we move the left pointer to the right by one



case: br

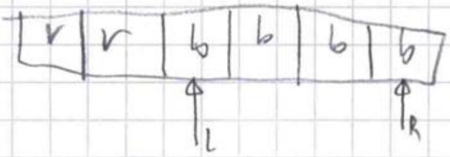
another case: br



repeat the above

Algorithm finishes in 2·n or O(n) time in the worst case (which is a single blue element in the beginning)

finally:



Ⓐ Ⓑ to Ⓒ Ⓓ

6. given an adjacency list Ⓐ

start at the first node
of the list and visit from Ⓑ
it's first neighbor by
adding to a stack Ⓒ
if it has not visited

Ⓓ

∅

then

No solution.
−10p

200828 Morgan Nyman
000302-5095 D 0041D

Page 9

7. $O(n)$ sort the integers using radix sort in $O(n)$ in the worst case.

$(0, n^2-1)$

How?

3

check for $y = c - x$
instead of $x + y = c$

```
if   y < c - x:  assign the next value in the list to y
if   y > c - x:       -||-   x
if   y == c - x: return true
else   return false
```

which list?