ds1pr21a3 Test 1

Due Mar 16, 2021 at 3:35pm **Points** 50 **Questions** 8
Available Mar 16, 2021 at 2pm - Mar 16, 2021 at 3:40pm about 2 hours
Time Limit 90 Minutes

Instructions

Many algorithms, methods, data structures and types have different versions. These exercises refer to the version you have seen in the classroom and/or you have learnt from my lecture notes (except of the structogram completing exercises).

This quiz is no longer available as the course has been concluded.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	88 minutes	34.52 out of 50

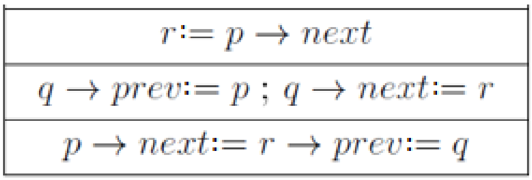
Correct answers are hidden.

Score for this quiz: **34.52** out of 50
Submitted Mar 16, 2021 at 3:28pm
This attempt took 88 minutes.

Question 1

2 / 2 pts

Which procedure body is given here?



```
$r := p \rightarrow next$
$q \rightarrow prev := p \; ; \; q \rightarrow next := r$
$p \rightarrow next := r \rightarrow prev := q$
```

○ unlink(q:E2*)

● follow(p,q:E2*)

○ precede(q,r:E2*)

Question 2

2 / 2 pts

Which operations of Q:Queue are correct?

☑ l:=Q.length()

□ Q.first(x)

□ x:=Q.top()

□ Q.isEmpty(b)

☑ x:=Q.rem()

□ Q.push(x)

Partial

Question 3

5 / 6 pts

Which of the following statements are True and which of them are False?

The answers must be given in T/ F form.

1. Merge sort is stable because there is not much difference between its best case and worst case performances. T

2. In quicksort, after partitioning the subarray before the pivot can be empty but the subarray after the pivot will never be empty. F

3. Merge sort is a a *divide and conquer* algorithm. T

4. The best case performance of merge sort and quicksort are asymptotically the same. T

5. Merge sort sorts in linear time in the best case. F

6. The worst case performance of insertion sort and quicksort are asymptotically the same. T

Answer 1:

T

Answer 2:

F

Answer 3:

T

Answer 4:

T

Answer 5:

F

Answer 6:

T

Partial

Question 4

6.86 / 8 pts

Perform insertion sort on the following array. A = [5, 2, 8, 3, 1, 2, 7, 5]

Give the content of the initial sorted subarray. Then give the sorted subarray after the given passes. The items must be separated by commas. No other character should be used, no blank. One pass is one iteration of the main loop of insertion sort.

Initial sorted subarray: A[0..0] = 5

The sorted subarray after the second pass: A[0..2] = 2,5,8

The sorted subarray after the fifth pass: A[0..5] = 1,2,2,3,5,8

Give the number of key comparisons and the number of data movements during the following passes.

During pass 2: key comparisons: 1 , data movements : 0

During pass 5: key comparisons: 4 , data movements : 3

Answer 1:

5

Answer 2:

2,5,8

Answer 3:

1,2,2,3,5,8

Answer 4:

1

Answer 5:

0

Answer 6:

4

Answer 7:

3

Question 5

8 / 8 pts

Perform merge sort on the following array. A = [6, 7, 8, 4, 3, 5, 2, 9, 1].

Give the result of each merge operation on the appropriate subarray in turn. The items must be separated by commas. No other character should be used, no blank.

1. 6,7

2. 4,8

3. 4,6,7,8

4. 3,5

5. 1,9

6. 1,2,9

7. 1,2,3,5,9

8. 1,2,3,4,5,6,7,8,9

Answer 1:

4,8

Answer 2:

4,6,7,8

Answer 3:

3,5

Answer 4:

1,9

Answer 5:

1,2,9

Answer 6:

1,2,3,5,9

Answer 7:

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

Partial

Question 6

2.67 / 8 pts

Sort array ⟨15,13,17,14,20,20,16,1⟩ with quicksort. We suppose that function partition() always selects the first item of the current subarray as pivot.

Give the subarrays computed by each partition(A, p, r) calls. The pivot must be distinguished by a '+' prefix. The items are separated by commas. No other character should be used, no blank.

For example: 4,2,+5,8

1,13,14+15,20,20,1

+1,13,14

+13,14

17,16,20,+20

16,+17

How many key comparisons were done during the quicksort?

14

Answer 1:

1,13,14+15,20,20,16,17

Answer 2:

+1,13,14

Answer 3:

+13,14

Answer 4:

17,16,20,+20

Answer 5:

16,+17

Answer 6:

14

Partial

Question 7

3 / 8 pts

Given a C2L with header. Pointer L refers to the header of the list. The keys of the items of the list are integers. The list is unsorted. Go through the list from beginning to end and delete the elements being less than their actual left neighbour. The resulting list will be sorted non-decreasingly.

Select the appropriate steps.

P:
[Select]

A:
[Select]

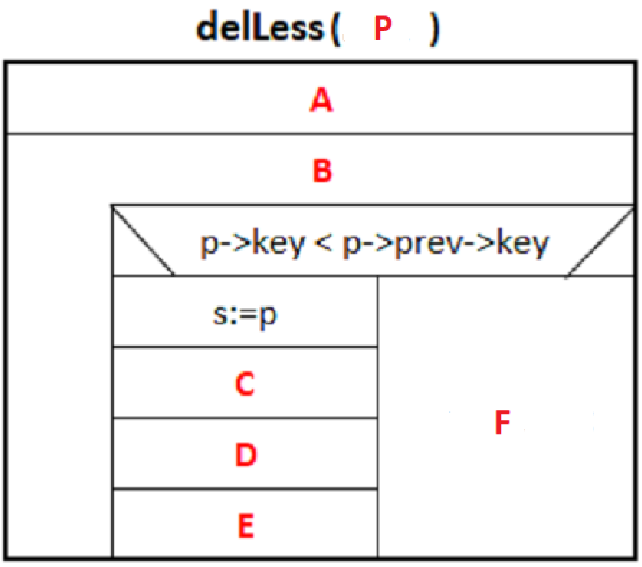
B:
[Select]

C:
[Select]

D:
[Select]

E:
[Select]

F:
[Select]



MT(n) ∈ [Select] where n is the original length of the list.

Answer 1:

L:E2*

Answer 2:

p:=L->next

Answer 3:

p ≠ L

Answer 4:

unlink(s)

Answer 5:

p:=p->next

Answer 6:

delete s

Answer 7:

p:=p->prev

Answer 8:

θ(n*n)

Partial

Question 8

5 / 8 pts

Pointer H refers to the header of a nonempty H1L. Relocate the items less than the first element of the list to the beginning of the list. The original order of the relocated items must not be changed. The elements are compared according to their keys.

Select the appropriate steps.

P: [Select] A: [Select] B: [Select]	

C:

[Select]

D:

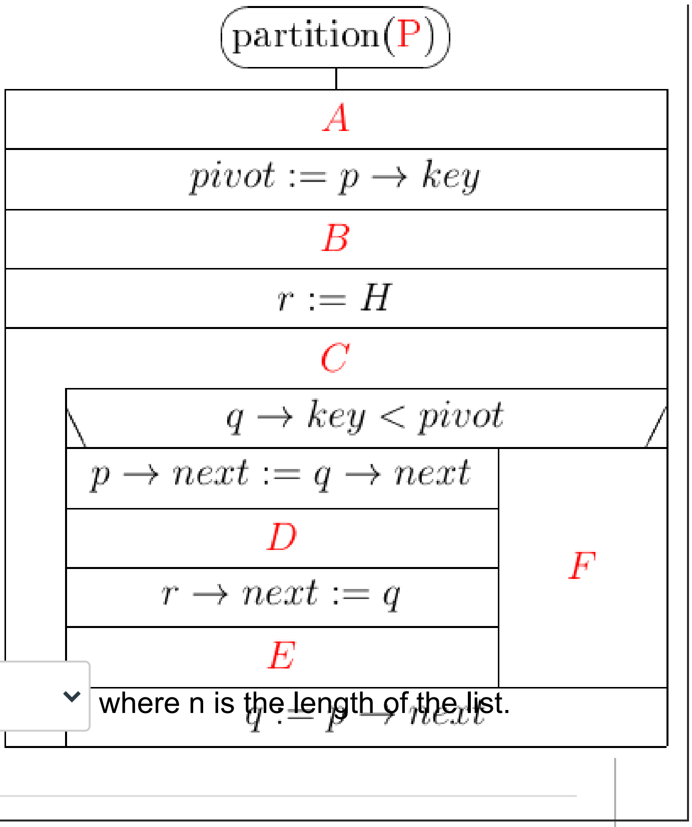
[Select]

E:

[Select]

F:

[Select]



MT(n) ∈ [Select] where n is the length of the list.

Answer 1:

H:E1*

Answer 2:

p:=H->next

Answer 3:

q:=p->next

Answer 4:

q ≠ 0

Answer 5:

q:=r->next

Answer 6:

p:=r

Answer 7:

p:=q

Answer 8:

θ(n*n)

Quiz Score: **34.52** out of 50