UnansweredQuestion 1

/ 1.5 pts

An ordinary, singularly-linked list can be traversed forward or backward.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 2

/ 1.5 pts

An ordinary, singularly-linked list allows individual links to be placed in any available place in the heap (free store).

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 3

/ 1.5 pts

An ordinary, singularly-linked list requires a contiguous block of memory to hold all the links.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 4

/ 1.5 pts

An ordinary, singularly-linked list has an insertion time of O(1) for inserting an item into the list.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 5

/ 1.5 pts

An ordinary, singularly-linked list has a deletion time of O(1) for finding and deleting an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 6

/ 1.5 pts

An ordinary, singularly-linked list has a replacement time of O(1) for finding and replacing an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 7

/ 1.5 pts

An ordinary, singularly-linked list has a retrieval time of O(1) for finding and returning at item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 8

/ 1.5 pts

An ordinary, singularly-linked list an easily and efficiently be used for a stack.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 9

/ 1.5 pts

An ordinary, singularly-linked list can easily and efficiently be used for a queue.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 10

/ 1.5 pts

An ordinary, singularly-linked list can be created (empty) in O(1) time.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 11

/ 1.5 pts

An ordinary, doubly-linked list can be traversed forward or backward.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 12

/ 1.5 pts

An ordinary, doubly-linked list allows individual links to be placed in any available place in the heap (free store).

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 13

/ 1.5 pts

An ordinary, doubly-linked list requires a contiguous block of memory to hold all the links.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 14

/ 1.5 pts

An ordinary, doubly-linked list has an insertion time of O(1) for inserting an item into the list.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 15

/ 1.5 pts

An ordinary, doubly-linked list has a deletion time of O(1) for finding and deleting an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 16

/ 1.5 pts

An ordinary, doubly-linked list has a replacement time of O(1) for finding and replacing an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 17

/ 1.5 pts

An ordinary, doubly-linked list has a retrieval time of O(1) for finding and returning at item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 18

/ 1.5 pts

An ordinary, doubly-linked list can easily and efficiently be used for a stack.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 19

/ 1.5 pts

An ordinary, doubly-linked list can easily and efficiently be used for a queue.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 20

/ 1.5 pts

An ordinary, doubly-linked list can be created (empty) in O(1) time.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 21

/ 1.5 pts

An open-indexing hash table uses separate chaining as an collision-resolution strategy.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 22

/ 1.5 pts

An open-indexing hash table allows individual buckets to be placed in any available place in the heap (free store).

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 23

/ 1.5 pts

An open-indexing hash table requires a contiguous block of memory to hold all the buckets.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 24

/ 1.5 pts

An open-indexing hash table has an insertion time of O(1) for inserting an item into the table.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 25

/ 1.5 pts

An open-indexing hash table has a deletion time of O(1) for finding and deleting an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 26

/ 1.5 pts

An open-indexing hash table has a replacement time of O(1) for finding and replacing an item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 27

/ 1.5 pts

An open-indexing hash table has a retrieval time of O(1) for finding and returning at item based on keys.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 28

/ 1.5 pts

An open-indexing hash table can easily and efficiently be used for a stack.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 29

/ 1.5 pts

An open-indexing hash table can easily and efficiently be used for a queue.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 30

/ 1.5 pts

An open-indexing hash table can be created (empty) in O(1) time.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 31

/ 1.5 pts

It usually takes fewer steps to insert an item in a hash table than in a linked list, as the number of items already contained in the data structure becomes large.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 32

/ 1.5 pts

It usually takes fewer steps to find a specific item in a hash table than in a linked list, as the number of items already contained in the data structure becomes large.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 33

/ 1.5 pts

It usually takes fewer steps to insert an item in a hash table than in an array, as the number of items already contained in the data structure becomes large.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 34

/ 1.5 pts

It usually takes fewer steps to find a specific item in a hash table than in an array, as the number of items already contained in the data structure becomes large.

Correct Answer

True

False

Additional Comments:

Question 35

/ 10 pts

Hashing

Given the following items to insert into a hash table of size 10, fill in the blanks/buckets in the table to show the hash table after all items have been inserted. If a blank/bucket should have no item in it after all items have been inserted into the table, put the word "none" in that location in the table.

The items are to be inserted starting from the top of the list and working down.

The primary hash function is key modulus table_size.

The collision resolution strategy is double hashing.

The secondary hash function is key div table_size, where div is integer division (that is, division discarding the remainder).

Items to insert

Item	Hash Code
A	54
B	43
C	28
D	60
E	33
F	79
G	81
H	41
I	88
J	67

Hash table

Bucket Number	Item
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Answer 1:

You Answered (You left this blank)

Correct Answer

D

Correct Answer

60

Correct Answer

d

Answer 2:

You Answered (You left this blank)

Correct Answer

G

Correct Answer

81

Correct Answer

g

Answer 3:

You Answered (You left this blank)

Correct Answer

I

Correct Answer

88

Correct Answer

i

Answer 4:

You Answered (You left this blank)

Correct Answer

B

Correct Answer

43

Correct Answer

b

Answer 5:

You Answered (You left this blank)

Correct Answer

A

Correct Answer

54

Correct Answer

a

Answer 6:

You Answered (You left this blank)

Correct Answer

H

Correct Answer

41

Correct Answer

h

Answer 7:

You Answered (You left this blank)

Correct Answer

E

Correct Answer

33

Correct Answer

e

Answer 8:

You Answered (You left this blank)

Correct Answer

J

Correct Answer

67

Correct Answer

j

Answer 9:

You Answered (You left this blank)

Correct Answer

C

Correct Answer

28

Correct Answer

c

Answer 10:

You Answered (You left this blank)

Correct Answer

F

Correct Answer

79

Correct Answer

f

Additional Comments:

Question 36

/ 20 pts

Linear Hashing

Given the following items to insert into a hash table that uses linear hashing, fill in the blanks/buckets in the table to show the hash table after all items have been inserted. To show how large the hash table has grown, only put bucket numbers in the blanks for the rows used in the table. For the remaining blanks, put the word "none" in place of the bucket number. Similarly, if a bucket or separate chain link should have no item in it after all items have been inserted into the table, put the word "none" in that location in the table.

The items are to be inserted starting from the top of the list and working down.

The collision resolution strategy is separate chaining.

Be sure to treat and write the hash codes and bucket numbers as binary numbers as shown in the 2018 slides.

Items to insert

Item	Hash Code

A	0010
B	1001
C	0000
D	1110
E	1000
F	0011
G	1100
H	1100
I	1101
J	1111

Hash table (with additional columns to represent where links for separate chaining would connect)

Bucket Number	Item	Link1	Link2

Answer 1:

You Answered (You left this blank)

Correct Answer

000

Answer 2:

You Answered (You left this blank)

Correct Answer

C

Correct Answer

c

Correct Answer

0000

Answer 3:

You Answered (You left this blank)

Correct Answer

E

Correct Answer

e

Correct Answer

1000

Answer 4:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 5:

You Answered (You left this blank)

Correct Answer

001

Answer 6:

You Answered (You left this blank)

Correct Answer

B

Correct Answer

b

Correct Answer

1001

Answer 7:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 8:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 9:

You Answered (You left this blank)

Correct Answer

010

Answer 10:

You Answered (You left this blank)

Correct Answer

A

Correct Answer

a

Correct Answer

0010

Answer 11:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 12:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 13:

You Answered (You left this blank)

Correct Answer

11

Answer 14:

You Answered (You left this blank)

Correct Answer

F

Correct Answer

f

Correct Answer

0011

Answer 15:

You Answered (You left this blank)

Correct Answer

I

Correct Answer

i

Correct Answer

1011

Answer 16:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 17:

You Answered (You left this blank)

Correct Answer

100

Answer 18:

You Answered (You left this blank)

Correct Answer

G

Correct Answer

g

Correct Answer

1100

Answer 19:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 20:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 21:

You Answered (You left this blank)

Correct Answer

101

Answer 22:

You Answered (You left this blank)

Correct Answer

H

Correct Answer

h

Correct Answer

1101

Answer 23:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 24:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 25:

You Answered (You left this blank)

Correct Answer

110

Answer 26:

You Answered (You left this blank)

Correct Answer

D

Correct Answer

d

Correct Answer

1110

Answer 27:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 28:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 29:

You Answered (You left this blank)

Correct Answer

111

Answer 30:

You Answered (You left this blank)

Correct Answer

J

Correct Answer

j

Correct Answer

1111

Answer 31:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 32:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 33:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 34:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 35:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 36:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 37:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 38:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 39:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 40:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Additional Comments:

Question 37

/ 30 pts

Radix Sort

```
// Radixsort takes:
// A: the array to sort
// r: the radix (base) for the keys to be sorted
// d: the number of digits (of the given radix) in each key
Algorithm Radixsort (A, r, d)
  create Q[ r ] // Q is an array of r queues, all initially empty
  for k from 0 to d-1
    for i from 0 to A.size
      Q[(A[ i ].key/(r to the power k)) modulus r].enqueue(A[ i ])
    end for i
    i ← 0
    for j from 0 to r do
      while Q[ j ] is not empty
        A[ i ] ← Q[ j ].dequeue()
        i ← i + 1
      end while
    end for j
  end for k
```

Given r is 10 and d is 2, show the steps followed by the Radix Sort algorithm given above in pseudocode when sorting the following array. Fill in the values in the figures for Q and A for each value of k. All empty locations should be marked "none" in these figures.

A (initially):

index	0	1	2	3	4	5	6	7	8	9
value	28	52	22	75	90	60	84	55	61	4

Q (when k = 0):

index	0	1	2	3	4	5	6	7	8	9
value at head										
value at next										

A (when k = 0):

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

Q (when k = 1):

index	0	1	2	3	4	5	6	7	8	9
value at head										
value at next										

A (when k = 1):

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

Answer 1:

You Answered (You left this blank)

Correct Answer

90

Answer 2:

You Answered (You left this blank)

Correct Answer

61

Answer 3:

You Answered (You left this blank)

Correct Answer

52

Answer 4:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 5:

You Answered (You left this blank)

Correct Answer

84

Answer 6:

You Answered (You left this blank)

Correct Answer

75

Answer 7:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 8:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 9:

You Answered (You left this blank)

Correct Answer

28

Answer 10:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 11:

You Answered (You left this blank)

Correct Answer

60

Answer 12:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 13:

You Answered (You left this blank)

Correct Answer

22

Answer 14:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 15:

You Answered (You left this blank)

Correct Answer

4

Correct Answer

04

Answer 16:

You Answered (You left this blank)

Correct Answer

55

Answer 17:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 18:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 19:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 20:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 21:

You Answered (You left this blank)

Correct Answer

90

Answer 22:

You Answered (You left this blank)

Correct Answer

60

Answer 23:

You Answered (You left this blank)

Correct Answer

61

Answer 24:

You Answered (You left this blank)

Correct Answer

52

Answer 25:

You Answered (You left this blank)

Correct Answer

22

Answer 26:

You Answered (You left this blank)

Correct Answer

84

Answer 27:

You Answered (You left this blank)

Correct Answer

75

Answer 28:

You Answered (You left this blank)

Correct Answer

55

Answer 29:

You Answered (You left this blank)

Correct Answer

28

Answer 30:

You Answered (You left this blank)

Correct Answer

4

Correct Answer

04

Answer 31:

You Answered (You left this blank)

Correct Answer

4

Correct Answer

04

Answer 32:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 33:

You Answered (You left this blank)

Correct Answer

22

Answer 34:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 35:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 36:

You Answered (You left this blank)

Correct Answer

52

Answer 37:

You Answered (You left this blank)

Correct Answer

60

Answer 38:

You Answered (You left this blank)

Correct Answer

75

Answer 39:

You Answered (You left this blank)

Correct Answer

84

Answer 40:

You Answered (You left this blank)

Correct Answer

90

Answer 41:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 42:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 43:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 44:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 45:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 46:

You Answered (You left this blank)

Correct Answer

52

Answer 47:

You Answered (You left this blank)

Correct Answer

61

Answer 48:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 49:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 50:

You Answered (You left this blank)

Correct Answer

none

Correct Answer

None

Correct Answer

"none"

Correct Answer

"None"

Answer 51:

You Answered (You left this blank)

Correct Answer

4

Correct Answer

04

Answer 52:

You Answered (You left this blank)

Correct Answer

22

Answer 53:

You Answered (You left this blank)

Correct Answer

28

Answer 54:

You Answered (You left this blank)

Correct Answer

52

Answer 55:

You Answered (You left this blank)

Correct Answer

55

Answer 56:

You Answered (You left this blank)

Correct Answer

60

Answer 57:

You Answered (You left this blank)

Correct Answer

61

Answer 58:

You Answered (You left this blank)

Correct Answer

75

Answer 59:

You Answered (You left this blank)

Correct Answer

84

Answer 60:

You Answered (You left this blank)

Correct Answer

90

Additional Comments:

Question 38

/ 6 pts

Match each term to a statement that is true for it. (Note that there are more statements than terms, so some statements will go unmatched.)

You Answered

Linear probing

resolves collisions by searching linearly through the hash table for an open bucket.

You Answered

Quadratic probing

resolves collisions by using increasingly large offsets.

You Answered

Double hashing

resolves collisions by using different offsets for different keys.

You Answered

A perfect hash function

never results in collisions.

You Answered

A minimal perfect hash function

never leaves empty buckets in the hash table.

You Answered

Separate chaining

resolves collisions through the use of external linked lists.

Other Incorrect Match Options:

performs in O(n log n) time.

resolves collisions through binary search.

is another name for folding.

resolves collisions using the quadratic formula.

performs in quadratic time, that is O(n^2) time.

resolves collisions through the use of a lookup table.

Additional Comments:

UnansweredQuestion 39

/ 1.5 pts

Radix sort is stable.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 40

/ 1.5 pts

Radix sort is an inplace algorithm.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 41

/ 1.5 pts

Radix sort manages to surpass O(n log n) performance on unique keys by not using key comparisons.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 42

/ 3 pts

Radix sort uses which of the following.

Doubly-linked lists

Hash tables

Binary search

Correct Answer

Queues

Stacks

Additional Comments:

UnansweredQuestion 43

/ 1.5 pts

Stacks are first-in/first-out (FIFO).

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 44

/ 1.5 pts

Queues are first-in/last-out (FILO).

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 45

/ 1.5 pts

Stacks can be built using linked lists.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 46

/ 1.5 pts

Queues can be built using linked lists.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 47

/ 1.5 pts

Stacks can be built using arrays.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 48

/ 1.5 pts

Queues can be built using arrays.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 49

/ 1.5 pts

Stacks allow for retrieval based on index.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 50

/ 1.5 pts

Queues allow for retrieval based on index.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 51

/ 1.5 pts

Primary clustering is a result of using prime numbers as hash table sizes.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 52

/ 1.5 pts

One advantage of using modulus arithmetic as a hash function is that it is fast to compute.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 53

/ 1.5 pts

Bucket search is the method used to find items in hash tables.

True

Correct Answer

False

Additional Comments:

UnansweredQuestion 54

/ 1.5 pts

Circular queues can be built using arrays.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 55

/ 1.5 pts

Circular queues can be built using linked lists.

Correct Answer

True

False

Additional Comments:

UnansweredQuestion 56

/ 3 pts

Adding n items to a linked list while keeping it sorted takes how much time?

LaTeX: \Theta\left(2^n\right) Θ (2 n)

Correct Answer

LaTeX: \Theta\left(n^2\right) Θ (n 2)

LaTeX: \Theta\left(n\right) Θ (n)

LaTeX: \Theta\left(n\:\log\:n\right) Θ (n log n)

LaTeX: \Theta\left(1\right) Θ (1)

Additional Comments:

UnansweredQuestion 57

/ 3 pts

Adding n items to an array while keeping it sorted takes how much time?

LaTeX: \Theta\left(2^n\right) Θ (2 n)

LaTeX: \Theta\left(1\right) Θ (1)

Correct Answer

LaTeX: \Theta\left(n^2\right) Θ (n 2)

LaTeX: \Theta\left(n\:\log\:n\right) Θ (n log n)

LaTeX: \Theta\left(n\right) $\Theta (n)$

Additional Comments: