1. [L.O.1.2] Indicate the complexity of the expression: n2 + 35n + 6

   A)O(n^3) B)O(n^2) C)O(n) D)O(42)

2. [L.O.1] Find the big-O of the following expression (the output of *f(n)*)

$$f(n) = \begin{cases} 2f(n-1) - 1, & n > 0 \\ 1, & n = 0 \end{cases}$$

   A)O(1) B)O(n) C)O(n^3) D)O(3^n)

3. [L.O.1.2] State the big-O of the following program:

```
i=1;
while(i<=n) i=i*3;
```

   A)O(n*3) B)O(n^3) C)O(i) D) Another answer

4. [L.O.1.1] Know that **Statement A** and **Statement B** both have O(1) complexity. Indicate the complexity of the following algorithm:

```
for(int i = 0; i<n; i++) Statement A;
for(int j = 0; j<m; j++) Statement B;
```

   A)O(n+m) B)O(max(n, m)) C)O(n*m) D)O(i+j)

5. [L.O.1.1] What is the complexity of the following code:

```
for(int i = 1; i<=n;i++)
      for(int j =1;j<=n;j++){
            cout << “*”;
            break;
      }
```

   A)O(n^2) B)O(n) C)O(1) D) Another answer

6. [L.O.1] Given the following program:

```
s=0;
for(int i=1; i<m; i++)
      for(int j=0; j<=i;j++)
            s+=j; //Line 1
```

   How many times will the source code “Line 1” be executed?

   A)(m+1)(m-1)/2 B)m(m-1)/2 C)(m+2)(m-1)/2 D)m(m+1)/2

7. [L.O.1.2] Given a function:

```
void func1(int arr[], int n){
    for(int i=n-1;i>=1;i--)
        for(int j=0; j<i;j++)
            if(arr[j]>arr[j+1])
                swap(arr[j], arr[j+1]);//Line 1
}
```

   In the worst case, how many times will the source code “Line 1” be executed?

   A)n(n-1)/2 B)n^2 C)n(n+1)/2 D) Another answer.

8. L.O.1] Given a function:

```
void my_recursive_function(int n){
     if(n == 0) return;
     my_recursive_function(n-1);
     cout << n << " ";
}
```

What is the output of calling function my_recursive_function(10)

A) Print out the number from 10 to 1 B) Print out the number from 10 to 0

C)Print out the number from 1 to 10 D) Print out the number from 0 to 10

9. [L.O.1] Indicates the execution result of the following program:

```
int fibo(int n){
      if(n == 1)          return 0;
      else if(n == 2)   return 1;
      return fibo(n - 1) + fibo(n - 2); }
int main() {
     int n = -1;
     int ans = fibo(n);
     cout << ans;
     return 0; }
```

A)0 B)1 C) Complier error D) Runtime error

10. [L.O.1.1] What is the screen output of the following program:

```
long int fun3(int x, int y){
    if(x>y)          return -1;
    else if(x==y)   return 1;
    else return x*fun3(x+1, y);
}
int main(){ cout << fun3(3, 7); return 0; }
```

A)360 B)400 C)300 D)320

11. [L.O.1.1] Which of the following statements is NOT TRUE about infinite recursion?

A) Infinite recursion will crash the program.

B) Infinite recursion consumes the entire system memory stack for the program and causes the program to terminate abnormally.

C) Calling indirect recursion always causes infinite recursion.

D) If the recursive call does not reach the base case, infinite recursion will occur.

12. [L.O.1.2] What is the output of the command: *func(301)* of the following function

```
void func(int number){
    if(number<=0) return;
    else{
        func(number/2);
        cout << number%2;
    }
}
```

A)100101101 B)101101001 C)100101010 D)101101010

13. [L.O.1.1] How many times the following fibo function is called when executing the below program?

```
int fibo(int n){
      if(n == 1) return 0;
      else if(n == 2) return 1;
      return fibo(n - 1) + fibo(n - 2);
}
```

```
int main(){
     int n = 5;
     int ans = fibo(n);
     return 0;
}
```

A)5 B)6 C)8 D)9

14. [L.O.1.2] Please indicate the screen output of the following program:

```
int cnt =0;
int foo(int n){
      if(n == 0)     return 0;
      return n % 10 + foo(n/10);
}
int my_function(int n, int sm){
      int i, tmp_sm;
      for(i=1;i<=n;i++){
          tmp_sm = foo(i);
          if(tmp_sm == sm)
            cnt++;
      }
      return cnt;
}

int main(){
      int n = 50, sum = 5;
      int ans = my_function(n,sum);
      cout << ans;
      return 0;
}
```

A)4 B)5 C)6 D)7

15. [L.O.2] Linked lists do NOT have any of the following characteristics
    A) When performing insertion and deletion operations, there is no need to move elements.
    B) There is no need to ask for memory allocation in advance.
    C) The speed of accessing elements is the same.
    D) The amount of memory allocated to a linked list is proportional (tỷ lệ thuận) to the length of the list.

16. [L.O.2] In a singly linked list, the time complexity of retrieving the previous and next elements of a certain element is:
    A)O(n) and O(1) B)O(1) and O(1) C)O(1) and O(n) D)O(n) and O(n)

17. [L.O.2] Suppose the most commonly used operation of a linked list is **to insert a new element at the end of the list** and **remove an element from the beginning of the list**. Please tell us which of the following data structures saves the most operation time.
    A) Singly linked lists use only the head pointer.
    B) Circular singly linked lists use only the head pointer.
    C) Doubly linked list.
    D) Circular singly linked lists use only tail pointers

18. [L.O.2] Given a circular singly linked list, with head and tail pointers. Indicate which of the following statements is correct?

A) tail→next == head B) tail → next →next == head
C) head →next == tail D) head →next →next == tail

19. [L.O.2.1] In a doubly linked list with a number of nodes greater than 10, suppose *p* points to a node in the list (this node is neither the first nor the last node). To insert node *q* before node *p*, which group of statements below is chosen?

A) p→previous = q; q→next = p; p→previous→next = q; q→previous = q;
B) p→previous = q; p→previous→next = q; q→next = p; q→previous = p→previous;
C) q→next = p; q→previous = p→previous; p→previous→next = q; p→previous = q;
D) q→previous = p→previous; q→next = q; p→previous = q; p→previous = q;

20. [L.O.2.2] Given a source code:

```
struct node{ int data; node* next;};
void rearrange(node *head) {
    node *p, *q;
    int temp;
    if (head == NULL || head->next == NULL) return;
    p = head;q = head->next;
    while (q != NULL) {
        temp = p->data; p->data = q->data;q->data = temp;
        p = q->next;
        q = p != NULL ? p->next : NULL;
    }
}
```

Know that the rearrange function receives the head parameter of a singly linked list: 1→2→3→4→5→6→7. What is the list after executing the rearrange(head) function call?

A) 1→2→3→4→5→6→7 B) 2→1→4→3→6→5→7
C) 1→3→2→5→4→7→6 D) 2→3→4→5→6→7→1

21. [L.O.2.2] Given a linked list using pointers head and tail. Which of the following tasks requires browsing the length of the list.

A)Remove the first node B)Swap data of the first and second node
C)Remove the last node D)Add a new node at the end of the list

22. [L.O.2] What is the value of the prefix: – * + 9 * 3 18 + 3 9 * 7 8 ?

A)500 B)600 C)700 D)800

23. [L.O.2] What is the value of the postfix: 1 2 + 3 * 15 3 / 6 + – 7 * ?

A)21 B)-21 C)14 D)-14

24. [L.O.2] The prefix form of the expression ((a + b*c)/(d +e))*(f – g) is:

A) * / + a * c b + d e – f g B) * / + a * b c + d e – f g
C) * / + a + b c * d e – f g D) * / + b * a c + d e – f g

25. [L.O.2] Given a non-empty non-full queue, which of the following task does not change the queue?

A)QueueFront/Top B)EnQueue/Add

C)DeQueue/Pop D)Another answer

26. [L.O.2.1] For a stack the add and remove of data following the principle:

A) First in first out B) Last in first out

C) Last in last out D) No principle

27. [L.O.2.1] Add n data a_1, a_2, a_3, …, a_n to a stack, then pop them out. If the first element out is a_n, What is the i_{th} ($1 \leq i \leq n$) element that pop out of the stack?

A) Unidentified B) a_{n-i+1} C) a_i D) a_{n-i}

28. [L.O.2.1] Using an array V[0..(n-1)] to implement a stack, when the stack is empty, the top of the stack top has value n, when adding x to the stack which of the following way is correct?

A)top:=top+1; V [top]:=x B) V [top]:=x; top:=top+1

C)top:=top-1; V [top]:=x D) V [top]:=x; top:=top-1

29. [L.O.2.1] An advantage of implement a stack using linked list compared to using array is:

A)Easier to add new elements

B)The stack rarely full

C)The stack never empty

D)Easier to remove elements.

30. [L.O.2.2] Using an array q[11] to implement a Circular Queue, the elements have the index from 0 to 10. Let the front and rear be 2. Then, add 9 elements, what is the value of front and rear.

A)front → 2, rear → 1; B)front → 2, rear → 0;

C)front → 1, rear → 0; C)front → 2, rear →10;

31. [L.O.2.2] Let IntQueue be an interger queue. What is the purpose of the following function:

```
void fun(int n){
    IntQueue q = new IntQueue();
    q.enqueue(0);
    q.enqueue(1);
    for (int i = 0; i < n; i++){
        int a = q.dequeue();
        int b = q.dequeue();
        q.enqueue(b);
        q.enqueue(a + b);
        print(a);
    }
}
```

A)Print from 0 to n-1

B)Print from n-1 to 0

C)Randomly print

D)Another answer

32. [L.O.2.1] In Bracket Matching problem, one checks the correctness of a sequence of parentheses. When solving this problem using stack with the input “[{}(){[]}]”, what is the maximum number of elements in the stack.

A)3 B)2 C)1 D)5

33. [L.O.3.3] Using Quicksort to sort the following numbers (28, 16, 32, 12, 60, 2, 5, 72) in ascending order. What is the result of the first Partition, Given the function Partition.

```
void Partition(int arr[], int low, int high){
    int left = low, right = high, pivot_item = arr[low];
      while(left <= right){
            while(arr[left] <= pivot_item) left++;
            while(arr[right] > pivot_item)  right--;
            if(left < right) swap(arr[left], arr[right]);
      }
      arr[low] = arr[right];
      arr[right] = pivot_item;
}
```

A)(2,5,12,16)28(60,32,72)
B)(5,16,2,12)28(60,32,72)
C)(2,16,5,12)28(60,32,72)
D)(5,16,12,2)28(32,60,72)

34. [L.O.3] Which of the following algorithms has the same complexity in all three cases (worst, average, best)

A) Straight insertion sort
B) Straight selection sort
C) Bubble sort
D) Shell sort

35. [L.O.3] Which of the following algorithms has the complexity of O(n) in the best case

A) Bubble sort
B) Straight selection sort
C) Mergesort
D) Another answer

36. [L.O.3] Using (straight insertion sort) to sort in ascending order. Which of following sequences of number has the least times of comparing?

A)94, 32, 40, 90, 80, 46, 21, 69
B)32, 40, 21, 46, 69, 94, 90, 80
C)21, 32, 46, 40, 80, 69, 90, 94
D)90, 69, 80, 46, 21, 32, 94, 40

37. [L.O.3.2] Given Straight insertion sort:

```
void insertionSort(int arr[], int array_size){
  int i, j, value;
  for (i = 1; i < array_size; i++){
          value = arr[i];
          j = i;
          while (________ ){
                   arr[j] = arr[j - 1];
                   j = j - 1;
          }
```

```
            arr[j] = value;
    }
}
```

Which of the following answer is correct fill of the blank (_________)?

A) (j > 0) || (arr[j − 1] > value) B) (j > 0) && (arr[j − 1] > value)

C) (j > 0) && (arr[j + 1] > value) D) (j > 0) && (arr[j + 1] < value)

38. [L.O.3.3] Given a 10000 elements array, if one wants to find 10 largest elements, which of the following algorithm has the best execution time?

A) Bubble sort B) Quicksort

C) Selection sort D) Insertion sort

39. [L.O.3.3] Sort the following array (81,94,11,96,12,35,17,95,28,58,41,75,15) using Shellsort. What is the result of one step (pass) with the value of increment 5.

A) 35,17,11,28,12,41,75,15,58,96,81,94,95

B) 35,17,11,41,12,12,75,15,96,58,81,94,95

C) 35,17,11,28,12,41,75,15,96,58,81,94,95

D) 35,17,11,12,28,41,75,15,96,58,81,94,95

40. [L.O.3.2] The complexity of mergesort is:

A)O(n) B)O(log(n)) C)$O(n^2)$ D)O(nlog(n))

------------- End of the Exam -------------

UNIVERSITY OF TECHNOLOGY (HCMUT) FACULTY OF CSE	MIDTERM EXAM	Semester/ Academic year	2	2023-2024
		Date	16/03/2024	
	Course title	Data Structures and Algorithms		
	Course ID	CO2003		
	Duration	60 mins	Code	3201

Notes: *- Closed book, do not use any course materials.*
- Submit the question sheet together with the answer sheet.
- Choose the best answer (only 1) for each question.

1. [L.O.1.2] Indicate the complexity of the expression: n2 + 35n + 6
   A)O(n^3) B)O(n^2) C)O(n) D)O(42)

2. [L.O.1] Find the big-O of the following expression (the output of *f(n)*)

$$f(n) = \begin{cases} 2f(n-1) - 1, & n > 0 \\ 1, & n = 0 \end{cases}$$

   A)O(1) B)O(n) C)O(n^3) D)O(3^n)

3. [L.O.1.2] State the big-O of the following program:

```
i=1;
while(i<=n) i=i*3;
```

   A)O(n*3) B)O(n^3) C)O(i) D) Another answer

4. [L.O.1.1] Know that **Statement A** and **Statement B** both have O(1) complexity. Indicate the complexity of the following algorithm:

```
for(int i = 0; i<n; i++) Statement A;
for(int j = 0; j<m; j++) Statement B;
```

   A)O(n+m) B)O(max(n, m)) C)O(n*m) D)O(i+j)

5. [L.O.1.1] What is the complexity of the following code:

```
for(int i = 1; i<=n;i++)
      for(int j =1;j<=n;j++){
            cout << “*”;
            break;
      }
```

   A)O(n^2) B)O(n) C)O(1) D) Another answer

6. [L.O.1] Given the following program:

```
s=0;
for(int i=1; i<m; i++)
      for(int j=0; j<=i;j++)
            s+=j; //Line 1
```

   How many times will the source code “Line 1” be executed?
   A)(m+1)(m-1)/2 B)m(m-1)/2 C)(m+2)(m-1)/2 D)m(m+1)/2

7. [L.O.1.2] Given a function:

```
void func1(int arr[], int n){
    for(int i=n-1;i>=1;i--)
```

```
          for(int j=0; j<i;j++)
              if(arr[j]>arr[j+1])
                  swap(arr[j], arr[j+1]);//Line 1
}
```

In the worst case, how many times will the source code “Line 1” be executed?

A)n(n-1)/2 B)n^2 C)n(n+1)/2 D) Another answer.

8. L.O.1] Given a function:

```
void my_recursive_function(int n){
     if(n == 0) return;
     my_recursive_function(n-1);
     cout << n << " ";
}
```

What is the output of calling function my_recursive_function(10)

A) Print out the number from 10 to 1 B) Print out the number from 10 to 0

C)Print out the number from 1 to 10 D) Print out the number from 0 to 10

9. [L.O.1] Indicates the execution result of the following program:

```
int fibo(int n){
      if(n == 1)        return 0;
      else if(n == 2)   return 1;
      return fibo(n - 1) + fibo(n - 2); }
int main() {
     int n = -1;
     int ans = fibo(n);
     cout << ans;
     return 0; }
```

A)0 B)1 C) Complier error D) Runtime error

10. [L.O.1.1] What is the screen output of the following program:

```
long int fun3(int x, int y){
    if(x>y)         return -1;
    else if(x==y)   return 1;
    else return x*fun3(x+1, y);
}
int main(){ cout << fun3(3, 7); return 0; }
```

A)360 B)400 C)300 D)320

11. [L.O.1.1] Which of the following statements is NOT TRUE about infinite recursion?

A) Infinite recursion will crash the program.

B) Infinite recursion consumes the entire system memory stack for the program and causes the program to terminate abnormally.

C) Calling indirect recursion always causes infinite recursion.

D) If the recursive call does not reach the base case, infinite recursion will occur.

12. [L.O.1.2] What is the output of the command: *func(301)* of the following function

```
void func(int number){
    if(number<=0) return;
    else{
        func(number/2);
        cout << number%2;
        }
    }
```

A)100101101 B)101101001 C)100101010 D)101101010

13. [L.O.1.1] How many times the following fibo function is called when executing the below program?

```
int fibo(int n){
      if(n == 1) return 0;
      else if(n == 2) return 1;
      return fibo(n - 1) + fibo(n - 2);
}
int main(){
     int n = 5;
     int ans = fibo(n);
     return 0;
}
```

A)5 B)6 C)8 D)9

14. [L.O.1.2] Please indicate the screen output of the following program:

```
int cnt =0;
int foo(int n){
      if(n == 0)    return 0;
      return n % 10 + foo(n/10);
}
int my_function(int n, int sm){
      int i, tmp_sm;
      for(i=1;i<=n;i++){
          tmp_sm = foo(i);
          if(tmp_sm == sm)
            cnt++;
      }
      return cnt;
}

int main(){
      int n = 50, sum = 5;
      int ans = my_function(n,sum);
      cout << ans;
      return 0;
}
```

A)4 B)5 C)6 D)7

15. [L.O.2] Linked lists do NOT have any of the following characteristics

A) When performing insertion and deletion operations, there is no need to move elements.

B) There is no need to ask for memory allocation in advance.

C) The speed of accessing elements is the same.

D) The amount of memory allocated to a linked list is proportional (tỷ lệ thuận) to the length of the list.

16. [L.O.2] In a singly linked list, the time complexity of retrieving the previous and next elements of a certain element is:

A)O(n) and O(1) B)O(1) and O(1) C)O(1) and O(n) D)O(n) and O(n)

17. [L.O.2] Suppose the most commonly used operation of a linked list is **to insert a new element at the end of the list** and **remove an element from the beginning of the list**. Please tell us which of the following data structures saves the most operation time.
A) Singly linked lists use only the head pointer.
B) Circular singly linked lists use only the head pointer.
C) Doubly linked list.
D) Circular singly linked lists use only tail pointers

18. [L.O.2] Given a circular singly linked list, with head and tail pointers. Indicate which of the following statements is correct?
A) tail→next == head B) tail → next →next == head
C) head →next == tail D) head →next →next == tail

19. [L.O.2.1] In a doubly linked list with a number of nodes greater than 10, suppose *p* points to a node in the list (this node is neither the first nor the last node). To insert node *q* before node *p*, which group of statements below is chosen?
A) p→previous = q; q→next = p; p→previous→next = q; q→previous = q;
B) p→previous = q; p→previous→next = q; q→next = p; q→previous = p→previous;
C) q→next = p; q→previous = p→previous; p→previous→next = q; p→previous = q;
D) q→previous = p→previous; q→next = q; p→previous = q; p→previous = q;

20. [L.O.2.2] Given a source code:

```
struct node{ int data; node* next;};
void rearrange(node *head) {
    node *p, *q;
    int temp;
    if (head == NULL || head->next == NULL) return;
    p = head;q = head->next;
    while (q != NULL) {
        temp = p->data; p->data = q->data;q->data = temp;
        p = q->next;
        q = p != NULL ? p->next : NULL;
    }
}
```

Know that the rearrange function receives the head parameter of a singly linked list: 1→2→3→4→5→6→7. What is the list after executing the rearrange(head) function call?
A) 1→2→3→4→5→6→7 B) 2→1→4→3→6→5→7
C) 1→3→2→5→4→7→6 D) 2→3→4→5→6→7→1

21. [L.O.2.2] Given a linked list using pointers head and tail. Which of the following tasks requires browsing the length of the list.
A)Remove the first node B)Swap data of the first and second node
C)Remove the last node D)Add a new node at the end of the list

22. [L.O.2] What is the value of the prefix: – * + 9 * 3 18 + 3 9 * 7 8 ?
A)500 B)600 C)700 D)800

23. [L.O.2] What is the value of the postfix: 1 2 + 3 * 15 3 / 6 + – 7 * ?

A)21 B)-21 C)14 D)-14

24. [L.O.2] The prefix form of the expression ((a + b*c)/(d +e))*(f – g) is:

A) * / + a * c b + d e – f g B) * / + a * b c + d e – f g

C) * / + a + b c * d e – f g D) * / + b * a c + d e – f g

25. [L.O.2] Given a non-empty non-full queue, which of the following task does not change the queue?

A)QueueFront/Top B)EnQueue/Add

C)DeQueue/Pop D)Another answer

26. [L.O.2.1] For a stack the add and remove of data following the principle:

A) First in first out B) Last in first out

C) Last in last out D) No principle

27. [L.O.2.1] Add n data a_1, a_2, a_3, …, a_n to a stack, then pop them out. If the first element out is a_n, What is the i_{th} (1 ≤ i ≤ n) element that pop out of the stack?

A) Unidentified B) a_{n-i+1} C) a_i D) a_{n-i}

28. [L.O.2.1] Using an array V[0..(n-1)] to implement a stack, when the stack is empty, the top of the stack top has value n, when adding x to the stack which of the following way is correct?

A)top:=top+1; V [top]:=x B) V [top]:=x; top:=top+1

C)top:=top-1; V [top]:=x D) V [top]:=x; top:=top-1

29. [L.O.2.1] An advantage of implement a stack using linked list compared to using array is:

A)Easier to add new elements

B)The stack rarely full

C)The stack never empty

D)Easier to remove elements.

30. [L.O.2.2] Using an array q[11] to implement a Circular Queue, the elements have the index from 0 to 10. Let the front and rear be 2. Then, add 9 elements, what is the value of front and rear.

A)front → 2, rear → 1; B)front → 2, rear → 0;

C)front → 1, rear → 0; C)front → 2, rear →10;

31. [L.O.2.2] Let IntQueue be an interger queue. What is the purpose of the following function:

```
void fun(int n){
    IntQueue q = new IntQueue();
    q.enqueue(0);
    q.enqueue(1);
    for (int i = 0; i < n; i++){
        int a = q.dequeue();
        int b = q.dequeue();
        q.enqueue(b);
        q.enqueue(a + b);
```

```
            print(a);
        }
    }
```

A)Print from 0 to n-1
B)Print from n-1 to 0
C)Randomly print
D)Another answer

32. [L.O.2.1] In Bracket Matching problem, one checks the correctness of a sequence of parentheses. When solving this problem using stack with the input "[{}(){[]}]", what is the maximum number of elements in the stack.
A)3 B)2 C)1 D)5

33. [L.O.3.3] Using Quicksort to sort the following numbers (28, 16, 32, 12, 60, 2, 5, 72) in ascending order. What is the result of the first Partition, Given the function Partition.

```
void Partition(int arr[], int low, int high){
    int left = low, right = high, pivot_item = arr[low];
      while(left <= right){
            while(arr[left] <= pivot_item) left++;
            while(arr[right] > pivot_item)  right--;
            if(left < right) swap(arr[left], arr[right]);
      }
      arr[low] = arr[right];
      arr[right] = pivot_item;
}
```

A)(2,5,12,16)28(60,32,72) B)(5,16,2,12)28(60,32,72)
C)(2,16,5,12)28(60,32,72) D)(5,16,12,2)28(32,60,72)

34. [L.O.3] Which of the following algorithms has the same complexity in all three cases (worst, average, best)
A) Straight insertion sort
B) Straight selection sort
C) Bubble sort
D) Shell sort

35. [L.O.3] Which of the following algorithms has the complexity of O(n) in the best case
A) Bubble sort
B) Straight selection sort
C) Mergesort
D) Another answer

36. [L.O.3] Using (straight insertion sort) to sort in ascending order. Which of following sequences of number has the least times of comparing?
A)94, 32, 40, 90, 80, 46, 21, 69 B)32, 40, 21, 46, 69, 94, 90, 80
C)21, 32, 46, 40, 80, 69, 90, 94 D)90, 69, 80, 46, 21, 32, 94, 40

37. [L.O.3.2] Given Straight insertion sort:

```
void insertionSort(int arr[], int array_size){
  int i, j, value;
  for (i = 1; i < array_size; i++){
          value = arr[i];
          j = i;
          while (________ ){
                   arr[j] = arr[j - 1];
                   j = j - 1;
          }
          arr[j] = value;
  }
}
```

Which of the following answer is correct fill of the blank (________)?

A) (j > 0) || (arr[j − 1] > value) [✓ B) (j > 0) && (arr[j − 1] > value)]

C) (j > 0) && (arr[j + 1] > value) D) (j > 0) && (arr[j + 1] < value)

38. [L.O.3.3] Given a 10000 elements array, if one wants to find 10 largest elements, which of the following algorithm has the best execution time?

A) Bubble sort B) Quicksort

[✓ C) Selection sort] D) Insertion sort

39. [L.O.3.3] Sort the following array (81,94,11,96,12,35,17,95,28,58,41,75,15) using Shellsort. What is the result of one step (pass) with the value of increment 5.

A) 35,17,11,28,12,41,75,15,58,96,81,94,95

B) 35,17,11,41,12,12,75,15,96,58,81,94,95

[✓ C) 35,17,11,28,12,41,75,15,96,58,81,94,95]

D) 35,17,11,12,28,41,75,15,96,58,81,94,95

40. [L.O.3.2] The complexity of mergesort is:

A)O(n) B)O(log(n)) C)$O(n^2)$ [✓ D)O(nlog(n))]

------------- End of the Exam -------------