Please use the “short” notation (don’t use constants). Example: `O(nk)` or `O(wn)` should be written `O(n)` . If an answer is required within a file, all your answers files must have a newline at the end.

Tests

Here is a quick tip to help you test your sorting algorithms with big sets of random integers: Random.org (/rltoken/YR-VWQbICB59wZs1eAal3w)

Quiz questions

Great! You've completed the quiz successfully! Keep going! (Hide quiz)

Question #0

What is the worst case time complexity of insertion in a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #1

What is the time complexity of accessing the nth element of a doubly linked list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #2

What is the time complexity of setting a value at index n in an unsorted array?

○ O(2^n)

(/)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #3

What is the time complexity of setting the value of the nth element in a singly linked list? (Assuming you have a pointer to the node to set the value of)

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #4

What is the time complexity of this function / algorithm?

```
void f(unsigned int n)
{
    int i;
    int j;

    for (i = 0; i < n; i++)
    {
        for (j = 1; j < n; j = j * 2)
        {
            printf("[%d] [%d]\n", i, j);
        }
    }
}
```

○ O(2^n)

● O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

(/)

○ O(log(n))

○ O(1)

Question #5

Assuming you have a pointer to the node to insert, what is the time complexity of inserting after the nth element of a doubly linked list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #6

What is the time complexity of worst case deletion from a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #7

What is the time complexity of this function / algorithm?

```
def func(n):
    a=5
    b=6
    c=10
    for i in range(n):
        for j in range(n):
            x = i * i
            y = j * j
            z = i * j
    for k in range(n):
        w = a*k + 45
        v = b*b
    d = 33
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

● O(n^2)

○ O(n)

○ O(log(n))

○ O(1)

Question #8

What is the time complexity of searching for an element in a stack of size n?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #9

Assuming you have a pointer to the node to set the value of, what is the time complexity of setting the value of the nth element in a doubly linked list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #10

What is the time complexity of the “push” operation onto a stack?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #11

What is the time complexity of accessing the nth element of a singly linked list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #12

What is the time complexity of inserting at index n on an unsorted array?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #13

(/)

What is the time complexity of this function / algorithm?

```
foreach($numbers as $number)
{
    echo $number;
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #14

Assuming you have a pointer to the node to remove, what is the time complexity of removing the nth element of a doubly linked list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #15

What is the time complexity of removing the nth element of a singly linked list? (Assuming you have a pointer to the node to remove)

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #16

What is the time complexity of this function / algorithm?

```
int Fibonacci(int number)
{
    if (number <= 1) return number;

    return Fibonacci(number - 2) + Fibonacci(number - 1);
}
```

● O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

○ O(1)

Question #17

What is the time complexity of removing at index n from an unsorted Python 3 list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #18

What is the time complexity of searching for an element in an unsorted Python 3 list of size n?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

(/)

Question #19

What is the time complexity accessing the nth element in an unsorted Python 3 list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #20

What is the time complexity of removing at index n in an unsorted array?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #21

What is the time complexity of searching for an element - worst case - in a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #22

What is the time complexity of accessing the nth element on an unsorted array?

○ O(2^n)

(/)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #23

What is the best case time complexity of insertion in a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #24

What is the time complexity of this function / algorithm?

```
void f(int n)
{
    int i;

    for (i = 0; i < n; i += 98)
    {
        printf("[%d]\n", i);
    }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

(/)

Question #25

What is the time complexity of setting value at index n in an unsorted Python 3 list?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #26

What is the time complexity of this function / algorithm?

```
var factorial = function(n) {
    if(n == 0) {
        return 1
    } else {
        return n * factorial(n - 1);
    }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #27

What is the time complexity of “popping” an element in a queue if you are given a pointer to both the head and the tail of the queue?

○ O(2^n)

○ O(log(n))

○ O(n!)

○ O(n)

○ O(nlog(n))

● O(1)
(/)

Question #28

What is the time complexity of best case deletion from a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #29

What is the time complexity of this function / algorithm?

```
void f(unsigned int n)
{
    int i;

    for (i = 1; i < n; i = i * 2)
    {
        printf("[%d]\n", i);
    }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

● O(log(n))

○ O(1)

Question #30

What is the time complexity of this function / algorithm?

```
void f(int n)
{
    int i;
    int j;

    for (i = 0; i < n; i++)
    {
        if (i % 2 == 0)
        {
            for (j = 1; j < n; j = j * 2)
            {
                printf("[%d] [%d]\n", i, j);
            }
        }
        else
        {
            for (j = 0; j < n; j = j + 2)
            {
                printf("[%d] [%d]\n", i, j);
            }
        }
    }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

● O(n^2)

○ O(n)

○ O(log(n))

○ O(1)

Question #31

What is the time complexity of searching for an element in an unsorted array of size n?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #32

What is the time complexity of inserting into an unsorted Python 3 list at index n?

(/)

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #33

What is the time complexity of searching for an element in a singly linked list of size n?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #34

What is the time complexity of searching for an element in a doubly linked list of size n?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #35

What is the time complexity of inserting after the nth element of a singly linked list? (Assuming you have a pointer to the node to insert)

○ O(2^n)

○ O(nlog(n))

○ O(n!)

(/)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Question #36

What is the time complexity of this function / algorithm?

```
void f(int n)
{
     int i;
     int j;

     for (i = 0; i < n; i++)
     {
          for (j = i + 1; j < n; j++)
          {
               printf("[%d] [%d]\n", i, j);
          }
     }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

● O(n^2)

○ O(n)

○ O(log(n))

○ O(1)

Question #37

What is the time complexity of searching for an element in a queue of size n if you are given a pointer to both the head and the tail of the queue?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)
(/)

Question #38

What is the time complexity of “pushing” an element into a queue if you are given a pointer to both the head and the tail of the queue?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #39

What is the time complexity of this function / algorithm?

```
void f(int n)
{
    printf("n = %d\n", n);
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #40

What is the time complexity of the “pop” operation onto a stack?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)
(/)

Question #41

What is the best case time complexity searching for an element in a hash table with the implementation you used during the previous Hash Table C project (chaining)?

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

○ O(n)

○ O(log(n))

● O(1)

Question #42

What is the time complexity of this function / algorithm?

```
void f(int n)
{
    int i;

    for (i = 0; i < n; i++)
    {
        printf("[%d]\n", i);
    }
}
```

○ O(2^n)

○ O(nlog(n))

○ O(n!)

○ O(n^2)

● O(n)

○ O(log(n))

○ O(1)

Tasks

0. Bubble sort

mandatory