Big O Quiz

Due No due date **Points** 10 **Questions** 10 **Time limit** None
Allowed attempts Unlimited

Instructions

No Marks! Just a practice.

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	2 minutes	10 out of 10
LATEST	Attempt 2	2 minutes	10 out of 10
	Attempt 1	49 minutes	5 out of 10

Submitted 19 Apr at 22:16

Question 1

1 / 1 pts

Which is the complexity for searching an unsorted list of length n?

○ O(n^2)

○ O(1)

○ O(logn)

○ O(mn), where m is the average size of the list elements

Correct!

● O(n)

If the list is unsorted, we need to do a exhaustive search. In the worst case, we will reach our target in the last element or the one we are looking for is not in the list.

Question 2

1 / 1 pts

We have an algorithm of O(n). Can we refer to this as O(n^2)? Pick the MOST CORRECT answer.

○ Yes, because it's still bound.

○ No, because O(n) isn't bounded by O(n^2) for some constants c>0.

Correct!

● Yes but O(n) is a tighter and better upper bound.

Usually, the bounds are the tighter the better.

○ No, because c.n^2 may be less than n for some constants c>0.

○ Yes, but only when n^2<n.

Question 3

1 / 1 pts

What is the complexity of two nested for loops, both over a range of 1...n?

○ Always O(1)

Correct!

● It depends on the statements inside the loop.

The total running time of the nested loops is the running time of the statements inside the loops multiplied by the product of the sizes of all the loops.

○ Always O(n).

○ Always O(nlogn)

○ Always O(n^2).

Question 4

1 / 1 pts

Suppose $T_1(n)$=O(f(n)) and $T_2(n)$ = O(f(n)). Which of the following are true?

○ T_1(n) = O(T_2(n))

○ T_1(n)/T_2(n) = O(1)

○ T_1(n)-T_2(n) = o(f(n))

Correct! ● T_1(n)+T_2(n) = O(f(n))

Question 5

1 / 1 pts

Please order the following functions by their order:

a. n log n

b. n log n^2

c. n $\log^2$ n

Correct! ● a = b <c

nlogn and nlogn^2 have the same order. n logn^2=2n logn = O(nlogn)

○ a < b < c

○ a = b =c

○ c < a < b

Question 6

1 / 1 pts

Which of the following options give a tight (can not be simply reduced) Big-Oh notation?

○ O(log n^3)

○ O(n^3 + log n)

○ O(n^2 + 3)

Correct! ● O(n log n)

Question 7

1 / 1 pts

What is the complexity of the following program fragments:

```
int sum = 0;

for(int i=1; i<n ; i++){

    for(int j=1; j<i*i; j++){

        for(int k=0; k<j; k++)

            sum++;

    }

}
```

○ O(n^4)

○ O(n^3)

Correct! ● O(n^5)

j can be as large as i^2, which can be as large as n^2.

k can be as large as j, which is n^2.

The running time of the nested loop depends on the size of the loop multiplied by the running time of the inside block, which is $O(n^5)$

○ O(n^2)

Question 8

1 / 1 pts

What is a better complexity of the following program fragments:

```
int sum = 0;
for(int i=1; i<n ; i++){
    for(int j=1; j<i*i; j++){
        if(j%i == 0)
            for(int k=0; k<j; k++)
                sum++;
    }
}
```

○ O(n^2)

○ O(n^3)

○ O(n^5)

○ O(n)

Correct!

● O(n^4)

The if statement is executed at most n^3 times. But the condition test only returns true for $O(n^2)$ times since it is true exactly i times for each i. Thus the innermost loop is only executed $O(n^2)$ times. Therefore the total running time is O(n^4)

Question 9

1 / 1 pts

Determine the O(g(n)) complexity of the following code.

Among choices below, choose the tightest g(n) that applies.

```
for (int i=0; i<n; i=i*2) {
  for(int j=0; j<i; j++) {
      cout << j;
  }
}
```

○ O(1)

Correct! ● O(n)

○ O(log(n))

○ O(n*log(n))

○ O(n^2)

The innermost loop performs 1+2+4+8+...+n/2+n=2n=O(n) operations overall. The overhead for the outermost loop is O(log(n)), which is less significant.

Question 10

1 / 1 pts

Determine the O(g(n)) complexity of the following code.

Among choices below, choose the tightest g(n) that applies.

```
int sum = 0;

for (int i=0; i<n; i++) {

  for(int j=n-1; j>=0; j--) {

     if (i == j) {

         for (int k=0; k<i; k++) {

              sum++;

         }

     }

  }

}
```

○ O(n^2*log(n))

○ O(1)

Correct! ● O(n^2)

○ O(n*log(n))

○ O(n^3)

○ O(n)

The innermost loop performs 1+2+3+...+ (n-1)=n(n-1)/2=O(n^2) operations, the overhead for outer loops is also O(n^2).