Q1) [Nested for loop] What is the time complexity of the following code snippet?

```
n = 10
m = 5
for i in range(n):
    for j in range(m):
        print((i, j))
```

A. O(n**2)
B. O(m**2)
C. O(m + n)
D. O(m*n)

Q2) [Two for loops] What is the time complexity of the following code snippet?

```
n = 10
m = 5
for i in range(n):
    print(i)

for j in range(m):
    print(j)
```

A. O(n**2)
B. O(m**2)
C. O(m + n)
D. O(m*n)

Q3) [Two while loops] What is the time complexity of the following code snippet?

```
n = 10
i = n
while i > 0:
    print(i)
    i -= 1

j = 1
while j < n:
    print(i + j)
    j *= 2
```

A. O(n**2)
B. O(log(n))
C. O(n * log(n))
D. O(n + log(n))

Q4) [Nested while loops - 1] What is the time complexity of the following code snippet?

```
n = 10
i = n

while i > 0:
    j = 0
    while j < n:
        print(i + j)
        j += 1
    i -= 1
```

A. O(n**2)
B. O(log(n))
C. O(n * log(n))
D. O(n + log(n))

Q5) [Nested while loops - 2] What is the time complexity of the following code snippet?

```
n = 10
i = n
j = 0

while i > 0:
    while j < n:
        print(i + j)
        j += 1
    i -= 1
```

A. O(n**2)
B. O(n)
C. O(n * log(n))
D. O(log(n))

Q6) [Nested while loops - 3] What is the time complexity of the following code snippet?

```
n = 10
i = n

while i > 0:
    j = 1
    while j < n:
        print(i + j)
        j *= 2
    i -= 1
```

A. O(n)
B. O(log(n))
C. O(n * log(n))
D. O(n + log(n))

Q7) [Nested while loops - 4] What is the time complexity of the following code snippet?

```
n = 10
i = n
j = 1

while i > 0:
    while j < n:
        print(i + j)
        j *= 2
    i -= 1
```

A. O(n)
B. O(log(n))
C. O(n * log(n))
D. O(n + log(n))

Q8) [Nested while loops - 5] What is the time complexity of the following code snippet?

```
n = 10
i = n
p = 0

while i > 0:
    p += 1
    j = 1
    while j < p:
        print(i + j)
        j *= 2
    i //= 2
```

A. O(log(log(n)))
B. O(log(n))
C. O(log(n) * log(n))
D. O(log(n) * log(log(n)))

Q9) [Nested while loops - 6] What is the time complexity of the following code snippet?

```
n = 10
i = n
p = 0

while i > 0:
    j = n
    while j > 0:
        print(i + j)
        j //= 2
    i //= 2
```

A. O(n**2)
B. O(log(n))
C. O(log(n) * log(n))
D. O(log(log(n)))

Q10) [Nested while loops - 7] What is the time complexity of the following code snippet?

```
import math
n = 256
j = 1

while j <= math.log(n, 2):
    print(j)
    j = j * 2
```

A. O(n)
B. O(log(n))
C. O(log(n) * log(n))
D. O(log(log(n)))

Solutions:

Q1) D. O(m * n)
Q2) C. O(m + n)
Q3) D. O(n + log(n))
Q4) A. O(n**2)
Q5) B. O(n)
Q6) C. O(n * log(n))
Q7) A. O(n)
Q8) D. O(log(n) * log(log(n)))
Q9) C. O(log(n) * log(n))
Q10) D. O(log(log(n)))