Select the appropriate asymptotic expression that bounds each sum defined below: * 6 points

Not all expressions will be used.

1. $1 + 2 + 3 + \cdots + N$
2. $2^1 + 2^2 + 2^3 + \cdots + N$
3. $3^1 + 3^2 + 3^3 + \cdots + 3^N$

	θ(logN)	θ(N)	θ(N²)	θ(2^N)	θ(3^N)
Row 1	○	○	○	○	○
Row 2	○	○	○	○	○
Row 3	○	○	○	○	○

Shawn sees the example in lecture of a function with amortized runtime and decides to write his own. He claims his implementation of addFirst() has a θ(1) amortized runtime. Is Shawn correct? * 2 points

```
public void addFirst(int x) {
    if (size == items.length) {
        resize(size + 1000);
    }
    items[size] = x;
    size += 1;
}
```

- Shawn is correct. Because the calls to resize() are very rare (1 out of every 1000 addFirst() calls), we can say that his addFirst() runs in θ(1) time on average.
- Shawn is not correct. For every N calls to addFirst, there will be N/1000 — or θ(N) — calls to resize(). Since each call to resize() takes linear time, the total cost of resizing after N addFirst's will be θ(N^2) (or θ(N) per call on average)

What is the runtime of the following method? * 1 point

```
public void eatThinMints(int n) {
  if (n == 0) return;

  System.out.println("om nom nom");

  for (int i = 0; i < 2; i++) {
    eatThinMints(n-1);
  }
}
```

- θ(logN)
- θ(N)
- θ(N^2)
- θ(2^N)

A copy of your responses will be emailed to yiyunchen@berkeley.edu.

Submit

Clear form