1 List’em all!

List all the asymptotic runtimes from quickest to slowest.
\( \theta(n^2), \theta(n^{0.5}), \theta(\log n), \theta(3^n), \theta(c), \theta(n^n), \theta(n\log n), \theta(n!), \theta(n^n), \theta(2^n) \)

2 What’s that runtime?

For each of the methods below, please specify the runtime in BigO, Big\( \Theta \) or Big\( \Omega \) Notation. Please give the tightest bound possible.

```java
private static void f(int n) {
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            linear(n); // runs in linear time with respect to input
        }
    }
}
```

```java
private static void g(int n) {
    if (n < 1) return;
    for(int i = 0; i < n; i++) {
        linear(100);
    }
    g(n/2);
    g(n/2);
}
```

```java
private static void h(int n) {
    Random generator = new Random();
    for(int i = 0; i < n; i++) {
        if(generator.nextBoolean()) {
            /* nextBoolean returns true with probability .5. */
            break;
        }
    }
}
```
private static void i(int n) {
    if (n < 1) return;
    for(int i = 0; i < n; i++) {
        System.out.println("Yow!");
    }
    i((999 * n) / 1000);
}

3 How fast?

Given an IntList of length N, provide the runtime bound for each operation. Recall that IntList is the naive linked list implementation from class.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>size()</td>
<td></td>
</tr>
<tr>
<td>get(int index)</td>
<td></td>
</tr>
<tr>
<td>addFirst(E e)</td>
<td></td>
</tr>
<tr>
<td>addLast(E e)</td>
<td></td>
</tr>
<tr>
<td>addBefore(E e, Node n)</td>
<td></td>
</tr>
<tr>
<td>remove(int index)</td>
<td></td>
</tr>
<tr>
<td>remove(Node n)</td>
<td></td>
</tr>
<tr>
<td>reverse()</td>
<td></td>
</tr>
</tbody>
</table>
4 Sum ’em Up

1. Define a function, sumTo, that takes a sorted int[] array and an int \( x \) and returns true if two numbers in the array sum to \( x \) and false otherwise. For example, if given the following input: \([1, 2, 4, 7, 8, 10]\) and \( x = 12 \), the function should return true.

2. Provide the tightest possible runtime bound on your solution.
5 Number Representation

Convert the following 4-bit numbers from signed integers to binary, and from binary to signed integers.

Decimal: 7 Binary: ____

Decimal: -5 Binary: ____

Decimal: __ Binary: 1000

Decimal: 3 + 7 Binary: ____

Now what is the decimal representation of this binary number? ____

Now for the questions below, consider that we are no longer working with 4-bit numbers, but rather 64 bit numbers.

Decimal: 1 << 2 Binary: ____

What is the decimal representation of this? ____

Decimal: 10 >> 2 Binary: ____

What is the decimal representation of this? ________

Given a number x, how do we determine if it’s even or odd using bit and boolean operators?

How do we determine whether x is a power of 2?

What is a number that can be represented as a 64 bit signed binary number but its absolute value cannot? (x can be represented but |x| cannot)