1 Algorithm Analysis

Some Useful Formulas:

\[
1 + 2 + 3 + 4 + \ldots + N = \frac{N(N + 1)}{2}
\]

\[
1 + 2 + 4 + 8 + \ldots + N = 2N - 1
\]

\[
2^0 + 2^1 + 2^2 + 2^3 + \ldots + 2^N = 2^{N+1} - 1
\]

\[
a + ar + ar^2 + \ldots + ar^{n-1} = \sum_{i=1}^{n} ar^{i-1} = a \frac{1 - r^n}{1 - r}
\]

Give the worst-case order of growth of the runtime in \(\Theta(\cdot)\) notation as a function of \(N\). Your answer should be simple with no unnecessary leading constants or summations.

(a) \(\Theta(____)\)

```java
static int recursion(int N) {
    int x = 0;
    if (N < 1000) {
        for (int i = 0; i <= N * N * N; i++) {
            x++;
        }
        return x;
    }
    for (int i = 1; i <= N / 2; i++) {
        x++;
    }
    return x + 2 * recursion(N / 2);
}
```

2 Array Resizing

We implement Queue ADT using ArrayQueue3 that add and remove methods work as expected. However, when we try to add a new element into the full Queue, we will make a new array with size of size+1 and copy every element over.

(a) What are the valid runtime bound of add in each case? (choose ALL that apply)

<table>
<thead>
<tr>
<th>Best case:</th>
<th>Worst Case:</th>
<th>Overall:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Omega(1))</td>
<td>(\Omega(1))</td>
<td>(\Omega(1))</td>
</tr>
<tr>
<td>(\Theta(1))</td>
<td>(\Theta(1))</td>
<td>(\Theta(1))</td>
</tr>
<tr>
<td>(O(1))</td>
<td>(O(1))</td>
<td>(O(1))</td>
</tr>
<tr>
<td>(\Omega(logN))</td>
<td>(\Theta(logN))</td>
<td>(\Theta(logN))</td>
</tr>
<tr>
<td>(\Theta(logN))</td>
<td>(O(logN))</td>
<td>(O(logN))</td>
</tr>
<tr>
<td>(O(N))</td>
<td>(\Theta(N))</td>
<td>(\Theta(N))</td>
</tr>
<tr>
<td>(\Omega(N))</td>
<td>(O(N))</td>
<td>(O(N))</td>
</tr>
<tr>
<td>(\Omega(N^2))</td>
<td>(\Theta(N^2))</td>
<td>(\Theta(N^2))</td>
</tr>
<tr>
<td>(\Omega(N^2))</td>
<td>(O(N^2))</td>
<td>(O(N^2))</td>
</tr>
</tbody>
</table>

(b) If ArrayQueue3 starts empty with an array of size 4, give a sequence of 6 operations that would perform the worst runtime (e.g., add/remove/add/remove/add/remove is a sequence of operations):

(c) What could be improved on ArrayQueue3 to improve overall performances? (one sentence should be enough.)
3 Algorithm Analysis (Optional)

(a) \( \Theta(\text{_____}) \)

static void addItUp(int N) {
  int x = 0;
  for (int i = 1; i <= N; i *= 2) {
    for (int j = 0; j < i; j++) {
      x++;
    }
    if (i % 2 == 1) {
      for (int j = 0; j < N; j++) {
        x++;
      }
    }
  }
  System.out.println(x);
}