1. Give a big-O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm.

```
$t := 0$
for $i := 1$ to $3$
  for $j := 1$ to $4$
    $t := t + ij$
```

2. Give a big-O estimate for the number additions used in this segment of an algorithm.

```
$t := 0$
for $i := 1$ to $n$
  for $j := 1$ to $n$
    $t := t + i + j$
```

3. Give a big-O estimate for the number of operations, where an operation is a comparison or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the **for** loops, where $a_1, a_2, ..., a_n$ are positive real numbers).

```
$m := 0$
for $i := 1$ to $n$
  for $j := i + 1$ to $n$
    $m := \max(a_i a_j, m)$
```

4. Give a big-O estimate for the number of operations, where an operation is an addition or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the **while** loop).

```
$i := 1$
$t := 0$
while $i \leq n$
    $t := t + i$
    $i := 2i$
```

6. a) Use pseudocode to describe the algorithm that puts the first four terms of a list of real numbers of arbitrary length in increasing order using the insertion sort.

b) Show that this algorithm has time complexity $O(1)$ in terms of the number of comparisons used.

***Note:** We will not ever ask you to write your own pseudocode.

9. Give a big-O estimate for the number of comparisons used by the algorithm that determines the number of 1s in a bit string by examining each bit of the string to determine whether it is a 1 bit (see Exercise 25 of Section 3.1).

11. a) Suppose we have n subsets $S_1, S_2, \dots, S_n$ of the set $\{1, 2, \dots, n\}$. Express a brute-force algorithm that determines whether there is a disjoint pair of these subsets. [*Hint:* The algorithm should loop through the subsets; for each subset S_i, it should then loop through all other subsets; and for each of these other subsets S_j, it should loop through all elements k in S_i to determine whether k also belongs to S_j.]

b) Give a big-O estimate for the number of times the algorithm needs to determine whether an integer is in one of the subsets.

***Note:** We will not ever ask you to write your own pseudocode.