Due: By 2pm on Friday 9 November. **Worth**: 5%

Keep in mind that your proofs will be graded on their structure at least as much as on their content. In other words, when we ask you to prove a statement, it is to find out how well you can write proofs, not because we are interested in knowing whether or not that statement is true. So pay particular attention to writing your proofs properly.

Consider the following algorithm.

```
MYSTERY($x, A, b, e$):
    if $e == b$:
        if $A[b] < x$:
            return 1
        else:
            return 0
    else:
        $c = \lfloor (2 \times b + e)/3 \rfloor$
        $d = \lfloor (b + 2 \times e)/3 \rfloor$
        return MYSTERY($x, A, b, c$) + MYSTERY($x, A, c+1, d$) + MYSTERY($x, A, d+1, e$)
```

In this problem set, we will consider only inputs x, A, b, e whose size $n = e - b + 1$ is a power of 3 (*i.e.*, $\exists k, n = 3^k$).

1. [5 marks]

   Write a recurrence relation for $T(n)$, the worst-case running time of algorithm MYSTERY on inputs of size $n = e - b + 1$, when n is a power of 3. Justify that your recurrence is correct from the algorithm—in particular, show your computation for the size of the input in each recursive call.

2. [10 marks]

   Find an exact closed-form expression for $T(n)$ when n is a power of 3 (show your work—using results proved in class is fine).

   Then, write a detailed proof that your closed form is correct for all n that are powers of 3.

3. [12 marks]

   State specific preconditions and postconditions for algorithm MYSTERY (remember to make your precondition as general as possible, and your postcondition as specific as possible).

   Then, prove that MYSTERY is correct with respect to your precondition and postcondition, for all inputs whose size n is a power of 3.

   WARNING: Your grade will depend heavily on having correct and reasonable preconditions and postconditions. In other words, you cannot "cheat" by stating a trivial postcondition that is very easy to prove—your postcondition must be a reasonable statement of the task carried out by the algorithm.

4. [3 marks]

   Is the algorithm correct for all inputs (even those whose size n is **not** a power of 3)? Justify your claim (*i.e.*, either give a counter-example or the main idea of a proof of correctness).