Q1 (10pt)

Let $S = x_1y_1 + x_2y_2 + \cdots + x_ny_n$, where $x_1, x_2, \cdots, x_n$ and $y_1, y_2, \cdots, y_n$ are orderings of two sequences of positive real numbers, each containing n elements.

a) Show that S takes its maximum value over all orderings of the two sequences when both sequences are sorted (so that elements in each sequences are in nondecreasing order).(5pt)

b) Show that S takes its minimum value over all orderings of the two sequences when one sequence is sorted into nondecreasing order and the other is sorted into nonincreasing order.(5pt)

Q2 (10pt)

Show that if a_n denotes the nth postive integer that is not a perfect square, then $a_n = n + \{\sqrt{n}\}$, where $\{x\}$ denotes the integer closed to the real number x. (An integer a is a perfect square if there is an integer b such that $a = b^2$.)

Q3 (10pt)

There exist mathematical functions that computers, no matter how powerful, simply cannot compute. We will do this through countability arguments.

a) Show that the set of *infinite* length binary strings is uncountable.(5pt)

b) A function is a *decision function* if it maps finite length bit strings into the range $\{0, 1\}$. Let F be the set consisting of *all possible* decision functions. Show that set F is uncountable.(5pt)

Q4 (10pt)

Analyze the average-case performance of the linear search algorithm, if exactly half the time the element x is not in the list, and if x is in the list, it is equally likely to be in any position.

Q5 (10pt)

a) Show that if f and g are real-valued functions such that $f(x)$ is $O(g(x))$, then for every positive integer n, $f^n(x)$ is $O(g^n(x))$. (Note that $fg : x \mapsto f(x)g(x)$.)(5pt)

b) Show that x^2 is $\mathcal{O}(x^4)$ but that x^4 is not $\mathcal{O}(x^2)$.(5pt)

Q6 (10pt)

Devise an algorithm that, given the binary expansions of the intergers a and b, determines whether $a > b$, $a = b$, or $a < b$.

Q7 (10pt)

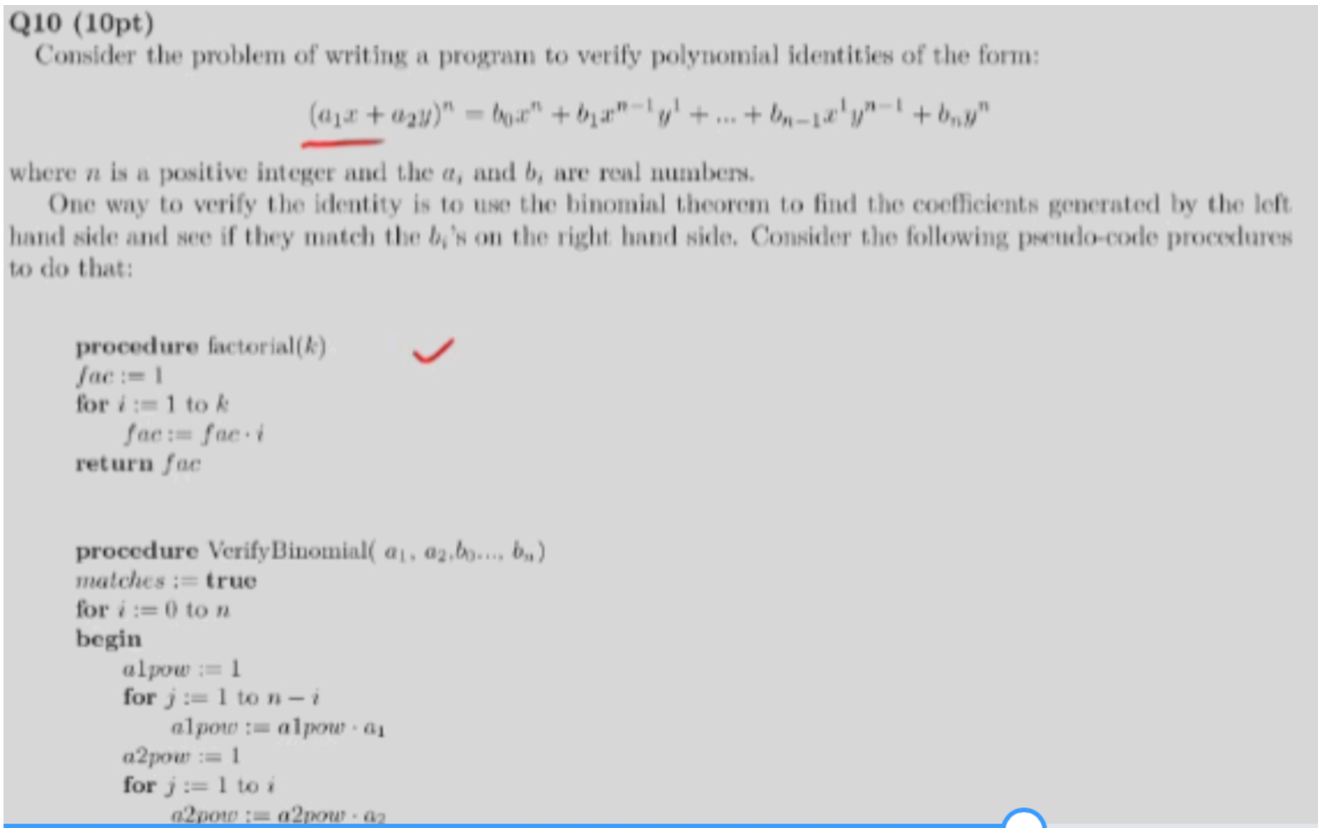
Find all solutions, if any, to the system of congruences $x \equiv 5 \pmod{6}$, $x \equiv 3 \pmod{10}$, and $x \equiv 8 \pmod{15}$.

Q8 (10pt)

Show that if p is prime, the only solutions of $x^2 \equiv 1 \pmod{p}$ are integers x such that $x \equiv 1 \pmod{p}$ or $x \equiv -1 \pmod{p}$.

Q9 (10pt)

Use mathematical induction to show that $\forall n \in \mathbb{N}$, 21 divides $4^{n+1} + 5^{2n-1}$.



Q10 (10pt)

Consider the problem of writing a program to verify polynomial identities of the form:

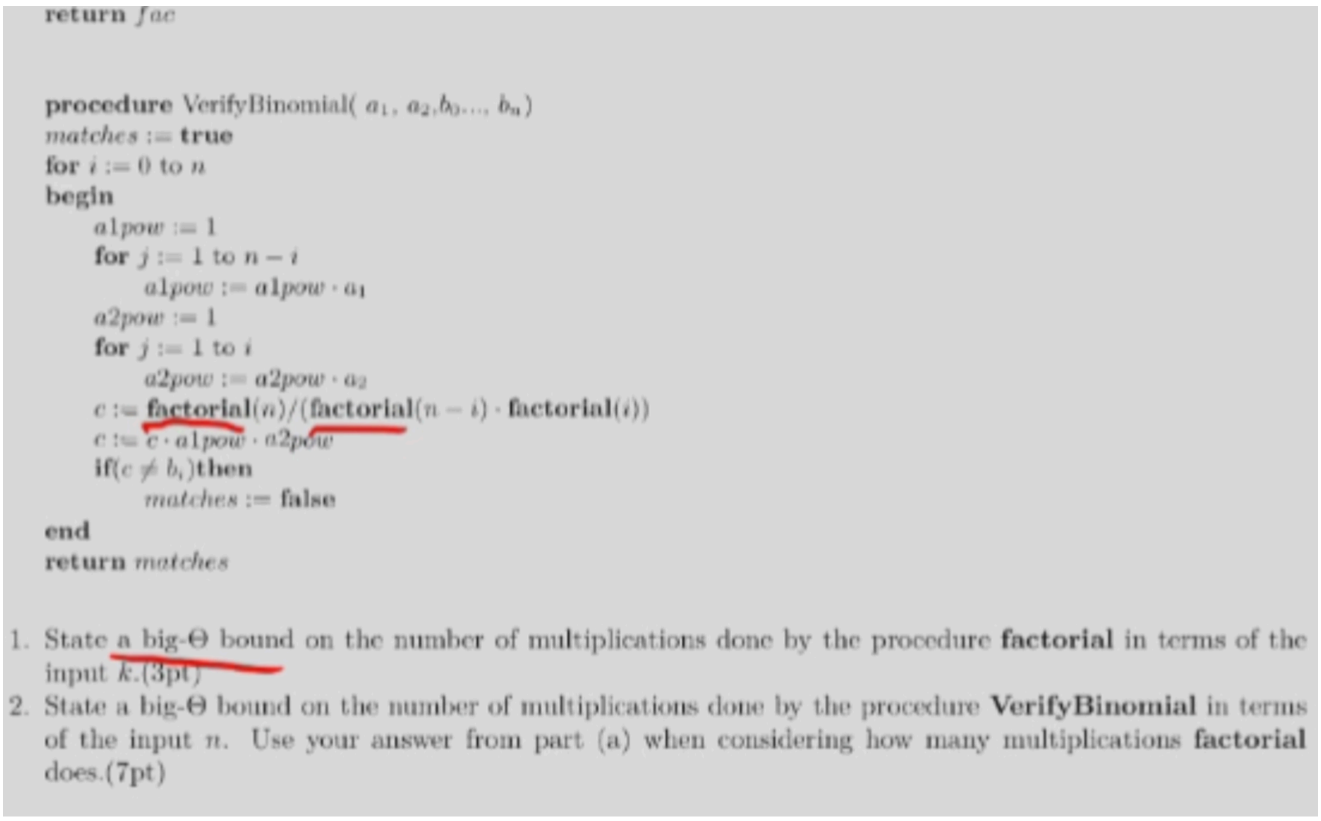
$$(a_1x + a_2y)^n = b_0x^n + b_1x^{n-1}y^1 + ... + b_{n-1}x^1y^{n-1} + b_ny^n$$

where n is a positive integer and the a_i and b_i are real numbers.

One way to verify the identity is to use the binomial theorem to find the coefficients generated by the left hand side and see if they match the b_i's on the right hand side. Consider the following pseudo-code procedures to do that:

```
procedure factorial($k$)
$fac := 1$
for $i := 1$ to $k$
    $fac := fac \cdot i$
return $fac$


procedure VerifyBinomial( $a_1$, $a_2$,$b_0$,..., $b_n$)
$matches :=$ true
for $i := 0$ to $n$
begin
    $a1pow := 1$
    for $j := 1$ to $n - i$
        $a1pow := a1pow \cdot a_1$
    $a2pow := 1$
    for $j := 1$ to $i$
        $a2pow := a2pow \cdot a_2$
```



```
return $fac$
```

```
procedure VerifyBinomial( $a_1$, $a_2$,$b_0$,..., $b_n$)
$matches :=$ true
for $i := 0$ to $n$
begin
    $a1pow := 1$
    for $j := 1$ to $n - i$
        $a1pow := a1pow \cdot a_1$
    $a2pow := 1$
    for $j := 1$ to $i$
        $a2pow := a2pow \cdot a_2$
```

```
    $c :=$ factorial($n$)/(factorial($n - i$) $\cdot$ factorial($i$))
    $c := c \cdot a1pow \cdot a2pow$
    if($c \neq b_i$)then
        $matches :=$ false
end
return $matches$
```

1. State a big-Θ bound on the number of multiplications done by the procedure **factorial** in terms of the input k.(3pt)
2. State a big-Θ bound on the number of multiplications done by the procedure **VerifyBinomial** in terms of the input n. Use your answer from part (a) when considering how many multiplications **factorial** does.(7pt)