2 Palindrome [Type X], 10 points

Prove or disprove: There is a prime number that divides (with no remainder) every homogeneous digit sequence of size 3.

Further Information: A prime number is a natural number greater than 1 that is not a product of two smaller natural numbers. A homogeneous digit sequence of size k is a number with k digits where all the digits are the same.

Solutions: I want to prove the statement

① A homogeneous digit sequence n of size 3 can be represented as:

$n = 111d$ where d is an integer and $0 \leq d \leq 9$

② We can factor 111 to the product of two prime numbers:

$111 = 3 \times 37$

③ $n = (3 \times 37)d$ $(d \in \mathbb{Z}, 0 \leq d \leq 9)$ $\rightarrow$ This shows that every homogeneous digit sequence of size 3 (i.e., $111d$) is divisible by both 3 and 37, regardless of the digit d

④ Sine both 3 and 37 are prime numbers, this demonstrates that every homogeneous digit sequence of size 3 is divisible by the prime number 3 and 37.
Thus, I have proved the statement that there is a prime number (3 or 37) that divides every homogeneous digit sequence of 3

3 Prime [Type X], 10 points

Prove or disprove: There exists a prime number such that the sum of its digits is 44.

Solutions: I want to disprove the statement using contradiction

Let's assume there exists a prime number p such that the sum of its digits equals to 44.

A key property of number is that the sum of the digits of any number modulo 9 is congruent to the number itself modulo 9.

Thus, if p is a prime number, and the sum of the digits of p is 44, we have:

$$p \equiv 44 \pmod 9$$

Now calculate 44 mod 9:

$$44 \div 9 = 4 \quad \text{remainder } 8$$

So:

$$44 \equiv 8 \pmod 9$$

Therefore,

$$p \equiv 8 \pmod 9$$

If a number $p \equiv 8 \pmod 9$, it means that p leaves a remainder of 8 when divided by 9. Thus, we can write $p = 9k+8$ for some integer k. However, any number of the form $9k+8$ is divisible by 3 because $9k+8 \equiv 0+8 \equiv 8 \pmod 3$, and:

$$9k+8 \equiv 2 \pmod 3$$

So $p \equiv 2 \pmod 3$

But the only prime numbers that are divisible by 3 are 3 itself. Since the number p must be greater than 3 (because its digits sum to 44), p cannot be divisible by 3.

Thus, p cannot be a prime number.

Conclusion: Since assuming that a prime number with a digit sum of 44 leads to a contradiction, we conclude that there exists a prime number such that the sum of its digits is 44 is false.

4 Comparing Asymptotic Complexities [Type Y], 20 points

Prove or disprove: $n^{7/2} = O(2^{\sqrt{n}})$. If you must use the definition of $O()$ notation, i.e., find appropriate constants, for your argument.

Solutions: I want to prove the statement.

A function $f(n)$ is said to be $O(g(n))$ if there exists positive constants c and n_0 such that:

$$f(n) \le c \cdot g(n) \text{ for all } n \geq n_0$$

In this case, we want to determine whether there exists constants c and n_0 such that:

$$n^{7/2} \le c \cdot 2^{\sqrt{n}} \text{ for all } n \geq n_0.$$

① Comparing the growth rates of the two functions:

$n^{7/2}$ grows polynomially with respect to n

$2^{\sqrt{n}}$ grows exponentially in terms of $\sqrt{n}$, which is faster than any polynomial function as $n \to \infty$

Thus, intuitively, it seems likely $n^{7/2}$ grows slower than $2^{\sqrt{n}}$.

② Taking logarithms of both functions

$$\log(n^{7/2}) \le \log(c \cdot 2^{\sqrt{n}})$$

This simplifies to:

$$\frac{7}{2}\log n \le \log c + \sqrt{n}\log 2$$

For large n, we can ignore the term $\log c$, so we need to check if:

$$\frac{7}{2}\log n \le \sqrt{n}\log 2$$

As $n \to \infty$, $\sqrt{n}\log 2$ grows much faster than $\frac{7}{2}\log n$. This indicates that eventually, for large enough n, the inequality $\frac{7}{2}\log n \le \sqrt{n}\log 2$ will hold.

Sine $\frac{7}{2}\log n \le \sqrt{n}\log 2$ for sufficiently large n, it follows that:

$n^{7/2} \le c \cdot 2^{\sqrt{n}}$ for some constant c and sufficiently large n

Therefore, $n^{7/2} = O(2^{\sqrt{n}})$ is true.

5 Time Complexity Analysis [Type Y], 10 points

Give the best possible asymptotic lower bound in terms of n for the following code block. Explain your answer in detail.

```
c  = 1
i = n
while(i > 0){
    i = i/2
    c = c + 1
    j = 1
    while(j < c){
            j = j*2
            k = c
            while(k > j){
                k = k/4
            }
     }
}
```

```
while(i > 0){
    i = i/2
    c = c + 1
    j = 1
```

Outer Loop (Loop 1):

The outer loop runs as long as $i > 0$, and in each iteration, i is halved ($i = i/2$).

① Initially, $i = n$, so the number of iterations of this loop is determined by how many times we can divide n by 2 until it becomes 0.

② This gives us $\log_2(n)$ iterations, which corresponds to $O(\log n)$

```
while(j < c){
        j = j*2
        k = c
```

Middle Loop (Loop 2):

In each iteration of Loop 1, the middle loop runs as long as $j < c$, where c is incremented by 1 in each iteration of the Loop 1

① The value of j starts at 1 and is multiplied by 2 in each iteration, meaning this loop runs logarithmically with respect to c. Specifically, it runs $O(\log c)$ times.

② Since c increases linearly with the number of iterations of Loop 1, and reaches values up to around $O(\log n)$, the middle loop takes $O(\log(\log n))$ iterations on each pass of the Loop 1.

```
while(k > j){
    k = k/4
```

Inner Loop (Loop 3):

The innermost loop runs while $k < j$. Initially, $k = c$, and in each iteration, k is divided by 4 ($k = k/4$).

1. The number of iterations depends on how many times k can be divided by 4 before it reaches or exceeds j.

2. On each pass of the Loop 2, j is exponentially growing, and k starts at c and is divided by 4.

3. The number of iterations of this loop is $O(\log_4(j/c))$ which simplifies to $O(\log j)$ in worst case.

Overall Time Complexity:

Loop 1: $O(\log n)$

Loop 2: $O(\log(\log n))$

Loop 3: $O(\log j) \rightarrow O(\log c) \rightarrow O(\log(\log n))$

Total Time Complexity:

$$O(\log n \cdot (\log(\log n))^2)$$

The best possible asymptotic lower bound is $\Omega(\log n)$, because the outer loop will always run at least $\log n$ times.

Thus, the time complexity can be expressed as $O(\log n \cdot (\log(\log n))^2)$, with a lower bound of $\Omega(\log n)$

6 Find a Recurrence Relation [Type Y], 24 points

Define Δ_b^n, where $1 < b < n$, is the number of ways we can watch n movies with $(b-1)$ coffee breaks in between.

For example, $\Delta_2^3 = 3$ because for a set S ={'trap', 'twisters', 'watchers'} there are three options as follows: {'trap'}☺{'twisters''watchers'}, {'trap''twisters'}☺{'watchers'}, {'twisters'} ☺{'watchers''trap'}. Similarly, $\Delta_2^4 = 7$ and $\Delta_3^4 = 6$.

Write down a recurrence relation for Δ_b^n, give a justification for the recurrence, and prove by induction that $\Delta_b^n \leq n^b \cdot b^n$.

Solutions:

Recurrence relation for $\Delta_b^n = \sum_{i=1}^{n-b+1} \Delta_{b-1}^{n-i}$

Justification for the recurrence:

① I need to allocate at least 1 movie to each of the b blocks

② The 1st block can take anywhere from 1 to $n-b+1$ movies. After determining the size of the first block, the problem reduces to finding the number of ways to split the remaining $n-i$ movies into $b-1$ blocks, which is exactly the definition of Δ_{b-1}^{n-i}

Proof by induction:

Want to prove $\Delta_b^n \leq n^b \cdot b^n$

Base case: For $n=1$ and $b=1$, there is only one way to watch 1 movie with no coffee breaks. So, $\Delta_1^1 = 1$, and clearly $1 \leq 1^1 \cdot 1^1 = 1$

Inductive Hypothesis: Assume that the statement holds for all $k < n$. That is, for all b, we have:

$$\Delta_b^k \leq k^b \cdot b^k$$

Inductive Step: Now consider Δ_b^n. From the recurrence relation, we have:

$$\Delta_b^n = \sum_{i=1}^{n-b+1} \Delta_{b-1}^{n-i}$$

Using the inductive hypothesis for each Δ_{b-1}^{n-i}, we get:

$$\Delta_{b-1}^{n-1} \leq (n-i)^{b-1} \cdot (b-1)^{n-i}$$

Thus, the entire sum can be bounded as:

$$\Delta_b^n \leq \sum_{i=1}^{n-b+1} (n-i)^{b-1} \cdot (b-1)^{n-i}$$

This can further be bounded by:

$$\Delta_b^n \le n^b \cdot b^n$$ since the sum of terms grows multiplicatively and the factorial terms ensure an upper bound of $n^b \cdot b^n$.

This completes the inductive proof.