CSSE1001 Week 11 Challenge Task — Solutions

1. Consider when `len(xs) == 1`, for example, `xs == [3]`. Then we will have `mid = 0`, and then `return add2([]) + add2([3])`. This leads to infinite computation, repeatedly calling `add2([3])`. It can also be shown that for `len(xs) >= 2`, we eventually arrive at the same problem.

   The modified solution is:

```
def add(xs):
    if xs == []:
        return 0
    elif len(xs) == 1:
        return xs[0]
    else:
        mid = len(xs) / 2
        return add(xs[:mid]) + add(xs[mid:])
```

2. Claim: `add(xs)` will always return the sum of `xs`.

   Proof. Let x_i denote `xs[i]`, and let $n =$ `len(xs)`. We must prove that `add(xs)` will always return $\sum_{i=0}^{n-1} x_i$. We proceed by induction on n:

   If $n = 0$, the `add(xs)` immediately returns $0 = \sum_{i=0}^{-1} x_i$, as required.

   Now, for $n \geq 1$, assume that the result holds for lists of length $n - 1$. Since `len(xs[1:])` $= n - 1$, we have that `add(xs[1:])` $= \sum_{i=1}^{n-1} x_i$. Then, `add(xs)` returns `xs[0] + add(xs[1:])`, which is

$$\texttt{add(xs)} = x_0 + \sum_{i=1}^{n-1} x_i = \sum_{i=0}^{n-1} x_i,$$

   as required. By induction, the result holds for any list `xs` of any length $n \geq 0$. □

3. Claim: `add2(xs)` will always return the sum of `xs`.

   Proof. Let x_i denote `xs[i]`, and let $n =$ `len(xs)`. We must prove that `add2(xs)` will always return $\sum_{i=0}^{n-1} x_i$. We proceed by strong induction on n:

   If $n = 0$ or $n = 1$, the `add(xs)` immediately returns the correct result.

   Now, assume that for some $n \geq 2$, the result holds for all $0 \leq k < n$. Then, if we let $m = \lfloor \frac{n}{2} \rfloor$, we have that $1 \leq m < n$. Therefore, `len(xs[:mid])` $= m < n$ and `len(xs[mid:])` $= n - m < n$, so by assumption, `add2(xs[:mid])` $= \sum_{i=0}^{m-1} x_i$, and `add2(xs[mid:])` $= \sum_{i=m}^{n-1} x_i$.

   Then, `add2(xs)` returns `add2(xs[:mid]) + add2(xs[mid:])`, which is

$$\texttt{add2(xs)} = \sum_{i=0}^{m-1} x_i + \sum_{i=m}^{n-1} x_i = \sum_{i=0}^{n-1} x_i,$$

   as required. By induction, the result holds for any list `xs` of any length $n \geq 0$. □

   Without the `elif len(xs) == 1` clause in the definition of `add2`, the inductive case of the above proof would have to cover $n \geq 1$ instead of $n \geq 2$. For $n = 1$ however, we do not have $1 \leq m$, so we cannot guarantee that `len(xs[mid:])` $< n$, and so we cannot apply the inductive hypothesis to `add(xs[mid:])`.