Python:

```python
def sum_strings(x, y):
    #reverse
    a, b = x[::-1], y[::-1]
    carry = 0
    result = []

    #essentially like how we learn  math
    for i in range(max(len(a), len(b))):
        digit_a = int(a[i]) if i < len(a) else 0
        digit_b = int(b[i]) if i < len(b) else 0

        total = digit_a + digit_b + carry
        carry = total // 10
        result.append(str(total % 10))

    # carry
    if carry != 0:
        result.append(str(carry))

    return ''.join(result[::-1]).lstrip('0') or '0'
```

sum_strings
Time Complexity:
The time complexity is O(N), where N is the length of the longer input string.
This is because the function iterates through each digit of the longer string once.

Space Complexity:
The space complexity is also O(N), as the function stores the result of each digit's sum in a list.
The length of this list is proportional to the length of the longer input string.

Python:

```python
def cakes(recipe, available):
    num_cakes_per_ingredient = []

    for ingredient, amount_needed in recipe.items():
        if ingredient not in available:
            return 0

        num_cakes = available[ingredient] // amount_needed
        num_cakes_per_ingredient.append(num_cakes)

    return min(num_cakes_per_ingredient)
```

cakes

Time Complexity:

The time complexity of the cakes function is O(M), where M is the number of ingredients in the recipe. This is because the function loops through each ingredient in the recipe exactly once.

Space Complexity:

The space complexity is O(M) as well, where M is the number of ingredients in the recipe. The function creates a list to store the number of cakes that can be made from each ingredient, and the size of this list is proportional to the number of ingredients.

Python:

```python
def make_readable(seconds):

    seconds = max(0, min(359999, seconds))
    hours = seconds // 3600
    minutes = (seconds % 3600) // 60
    seconds = seconds % 60

    return "{:02d}:{:02d}:{:02d}".format(hours, minutes, seconds)
```

make_readable

Time Complexity:

The time complexity of the function is O(1). This is because the function performs a fixed number of operations regardless of the input size.

Space Complexity:

The space complexity is also O(1). The function uses a fixed amount of space for the variables hours, minutes, and seconds, and the space required for the output string is also constant, as it always consists of 8 characters.