Please provide the Big-O analysis for each of the algorithms below:

1. Add two numbers:

```
def add(x, y):
    return x + y
```

Time complexity:

Space complexity:

---

2. Search in a list

```
def find(arr, target):
    for item in arr:
        if item == target:
            return True
    return False
```

Time complexity:

Space complexity:

---

3. Search in a nxn matrix

```
def find(matrix, target):
    if len(matrix) > 0:
        for i in range(len(matrix)):
            for j in range(len(matrix[0])):
                if matrix[i][j] == target:
                    return True
    return False
```

Time complexity:

Space complexity:

4. Remove a given character from string (for example, remove all commas)

```
def replace(word, char_to_remove, char_to_insert=''):
    temp = []
    for c in word:
        if c != char_to_remove:
            temp.append(c)
        else:
            temp.append(char_to_insert)
    return ''.join(temp)
```

Time complexity:

Space complexity:

---

5. Apply lambda function to dataframe

```
df['new_column'] = df['old_column'].apply(lambda x: x * 1000)
```

Time complexity:

Space complexity:

---

6. Bubble Sort

```
def bubble_sort(arr):
    sorted = False
    while(not sorted):
        sorted = True
        for i, value in enumerate(arr[:-1]):
            if value > arr[i+1]:
                arr[i], arr[i+1] = arr[i+1], arr[i]
                sorted = False
    return arr
```

Time complexity:

Space complexity:

Challenges

7. Find all contiguous substrings

```
def find_cont_substrings(arr):
    substrings = set()
    for i in range(len(arr)):
        for j in range(i,len(arr)):
            substrings.add(tuple(arr[i:j+1]))
    return substrings
```

Or, using list comprehension:

```
def find_substrings(arr):
    length = len(arr)
    return [arr[i:j+1] for i in range(length) for j in range(i,length)]
```

Time complexity:

Space complexity:

---

8. Find or Implement a sorting method faster than Bubble-Sort

Time complexity:

Space complexity:

---

9. Generate the power set of a list

Time complexity:

Space complexity: