QUESTION 1. [12 marks]
I have left out the documentation for function \( s(n) \). Work out what it produces, starting at the smallest \( n \) and working up.

```python
def s(n: int) -> tuple:
    """For you to figure out...""
    if n == 1:
        return (1, 1)
    else:
        return min([(2 * s(n - i)[0] + 2**i - 1, i) for i in range(1, n)])
```

**Part (a)** [2 marks]
s(1): (1, 1)

**Part (b)** [2 marks]
s(2): (3, 1)

**Part (c)** [2 marks]
s(3): (5, 2)

**Part (d)** [2 marks]
s(4): (9, 2)

**Part (e)** [2 marks]
s(5): (13, 2)

**Part (f)** [2 marks]
s(6): (17, 3)

QUESTION 2. [15 marks]
Python's `str.join` concatenates a list of strings, using the given separator string, and returns the new string, as follows:

```python
>>> str.join('+', ['one', 'two', 'three'])
'one+two+three'
```

Complete the definition of `nested_join(...)`, which concatenates the strings in a nested list of strings, using the given separator string, and returns the result.

```python
def nested_join(s: str, L: list) -> str:
    """Return join of nested list of strings L with separator string s"

    >>> nested_join(' ', [])
    ''
```

Page 1 of 4
CONT'D...
>>> nested_join(' ', ['one'])
'one'
>>> nested_join(' ', ['one', 'two'])
'one two'
>>> nested_join(' ', ['one', ['two', 'three'], 'four'])
'one two three four'

SOLUTION:

return str.join(s, [nested_join(s, x) if isinstance(x, list) else x for x in L])

QUESTION 3.  [15 MARKS]

You are to implement FunctionalList, a subclass of built-in class list. FunctionalList should have two new methods:

functional_append(self: 'FunctionalList', o: object) -> 'FunctionalList':

"""Return a copy of this FunctionalList with o appended""

def functional_append(self: 'FunctionalList', o: object) -> 'FunctionalList':
    return FunctionalList([x for x in self] + [o])

def functional_sort(self: 'FunctionalList') -> 'FunctionalList':
    """Return a sorted copy of this FunctionalList""
    L = [x for x in self]
    L.sort()
    return FunctionalList(L)

PART (A)  [8 MARKS]

Implement FunctionalList in the space below.
PART (B)  [7 marks]

Describe three more test cases that would increase your confidence that functional_append and functional_sort work as specified. One example is given.

- Create FL = FunctionalList([1]). Then FL.functional_append(2) returns the FunctionalList [1, 2] and FL is left unchanged.

- Create FL = FunctionalList([]). Then FL.functional_append(2) returns the FunctionalList [2] and FL is left unchanged.

- Create FL = FunctionalList([3, 1, 2]). Then FL.functional_sort() returns the FunctionalList [1, 2, 3] and FL is left unchanged.

- Create FL = FunctionalList([3]). Then FL.functional_append(2).functional_sort() returns the FunctionalList [2, 3] and FL remains unchanged.
This page is left (mainly) blank for things that don’t fit elsewhere.

# 1: _____/12
# 2: _____/15
# 3: _____/15

TOTAL: _____/42