Assignment 4- Sarah Sindeband

1. Describe a recursive algorithm for finding the maximum element in a sequence, S, of n elements.

```
def max_search(S):
    n = len(S) # n elements in sequence S
    count = 0
    if n == count: # if count has reached the number of elements
        return False # stop the loop
    else:
        large = search_max(S[count+1:]) # calling the function again for index count +1 to the end
        if(large < S[count]): # if index count is greater than the numbers in the remaining index
            large = S[count] # base case, large becomes index count
        count +=1 # update count
        return large
```

   What is your running time and space usage?

   The recursive functions have a characteristic of O(n) for the linear running time and space useage because the function will be called n times.

2. Write a short recursive Python function that finds the minimum and maximum values in a sequence without using any loops.

```
def find_max(S, n)
    if(n == 1):
        return S[0]
    else:
        return max(S[n-1], find_max(S, n-1))
```

3. Draw the recursion trace for this function for a sequence of 6 elements. A recursion trace skeleton is provided below.

   S = [9, 8, 7, 6, 5, 4]