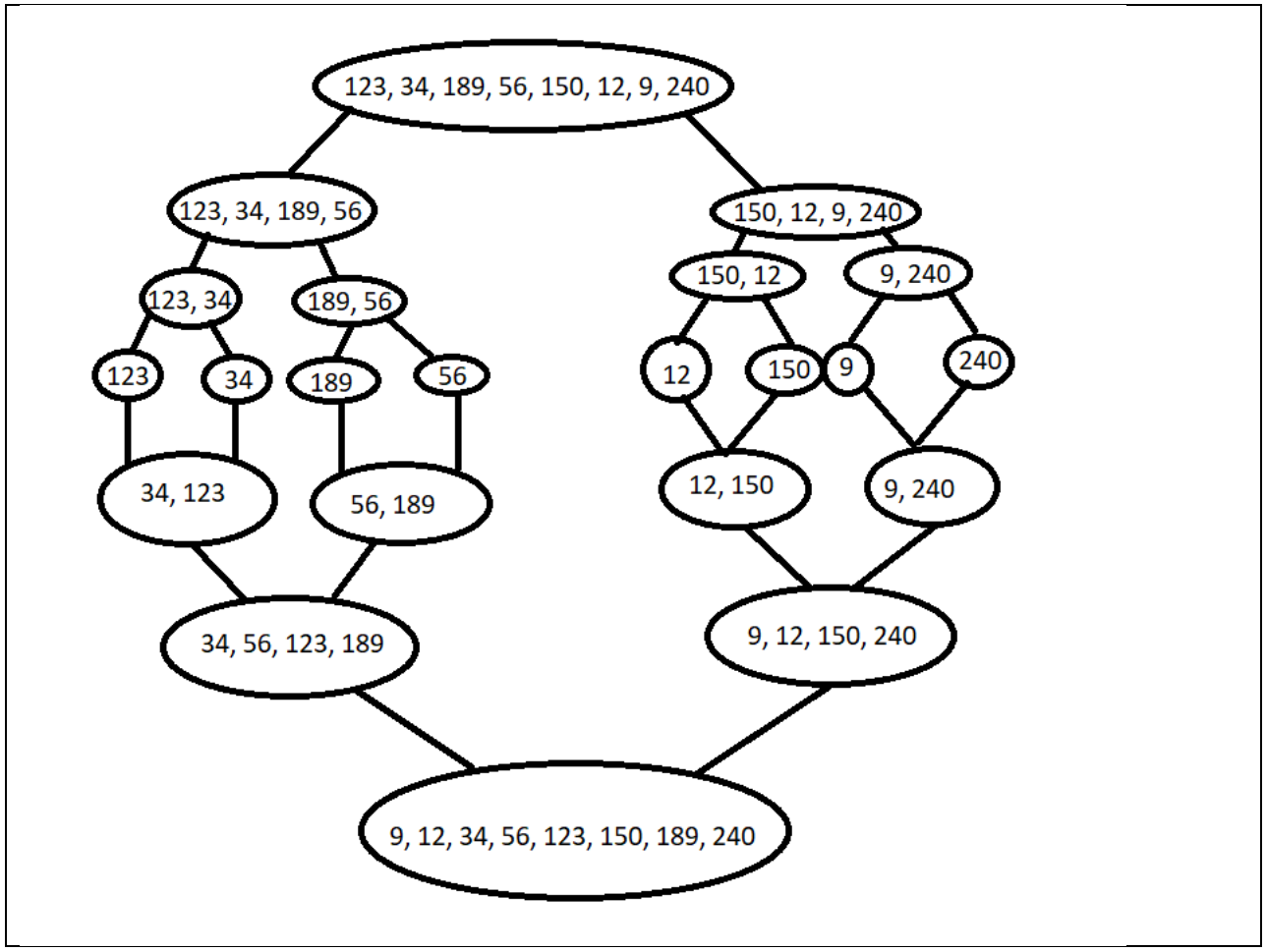
Brendan Morgenstern

CUS 715

Problem Set 4

Merge sort will divide a single array into 2 subarrays of equal or roughly equal lengths. In the case of the array [123, 34, 189, 56, 150, 12, 9, 240], it will become [123, 34, 189, 56] and [150, 12, 9, 240]. These sub arrays will also recursively be split until the arrays are of length one. The separate length 1 arrays are merged back into a single array until all of the subarrays have been merged. This diagram demonstrates how the array is split and later merged back.



When merge sort is actually performed with debugging info set, the steps are given explicitly.

```
C:\Users\Commander Shepard\Desktop\School\Algorithms\MergeSort>mtest.py 2 debug
Split into [123, 34, 189, 56]
Split into [123, 34]
Split into [123]
Split into [34]
34 taken from right over 123
Split into [189, 56]
Split into [189]
Split into [56]
56 taken from right over 189
34 taken from left over 56
56 taken from right over 123
123 taken from left over 189
Split into [150, 12, 9, 240]
Split into [150, 12]
Split into [150]
Split into [12]
12 taken from right over 150
Split into [9, 240]
Split into [9]
Split into [240]
9 taken from left over 240
9 taken from right over 12
12 taken from left over 240
150 taken from left over 240
9 taken from right over 34
12 taken from right over 34
34 taken from left over 150
56 taken from left over 150
123 taken from left over 150
150 taken from right over 189
189 taken from left over 240
[9, 12, 34, 56, 123, 150, 189, 240]
```

Traditionally, Merge Sort splits the array to be sorted into 2 parts at a time, though this can be expanded to split it into k parts where k is a real number greater than 1. The code for my implementation can be found in my publicly available Github repo here, but the main part of the code that implements this “by k” split is as follows.

```
def splitbyn(listvar, parts):
    newlist = []
    midpoint = int(len(listvar)/parts)
    if int(len(listvar)%parts) >= int(parts/2):
        midpoint += 1
    pos = 0
    for i in range(1,parts):
        nextsl = pos+midpoint
        newlist.append(listvar[pos:nextsl])
        pos = nextsl
    newlist.append(listvar[pos:])
    return newlist


def MergeSort(src_array, splitby=2, debug=False):
    sorted = []
    if len(src_array) < 2:
        return src_array
    elif splitby < 2:
        print("Cannot split into less than 2 parts")
        return src_array
    for subarray in splitbyn(src_array, splitby):
        if debug:
            print("Split into "+ str(subarray))
        sorted = Merge(sorted,MergeSort(subarray,splitby,debug),debug)

    return sorted
```