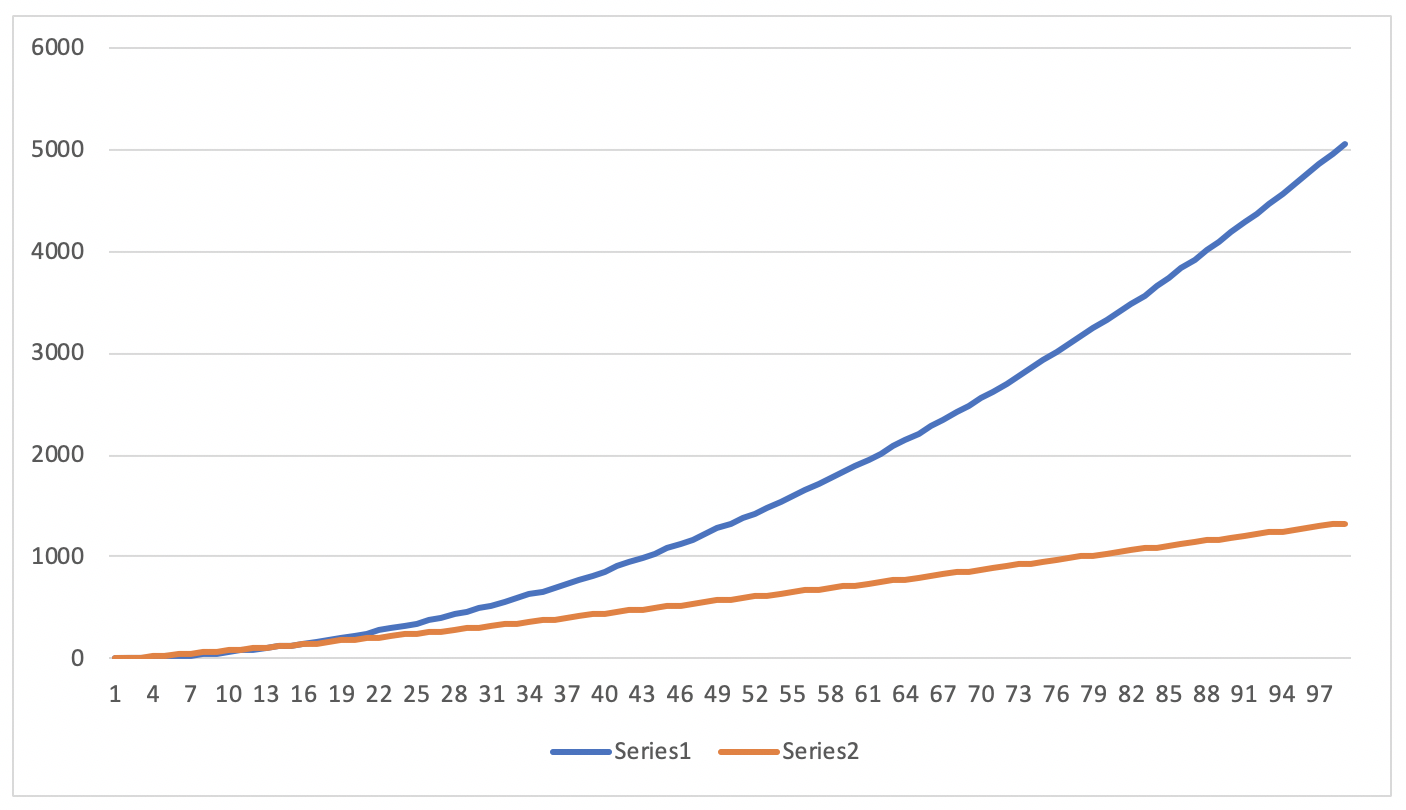
**Selection Sort** (Blue line) treats the input as two parts, a sorted part and an unsorted part, and repeatedly selects the next value to move from the unsorted part to the end of the sorted part. The unsorted part is searched to find the index of the element with the smallest value. The variable indexSmallest stores the index of the smallest element in the unsorted part. Once the element with the smallest value is found, that element is swapped with the element at that location. It may require a large number of comparisons as you can see on the graph. The runtime is O(N). If a list has N elements, the outer loop executes N - 1 times. For each of those N - 1 outer loop executions, the inner loop executes an average of times. Thus, the total number of comparisons is proportional to, or O(N).

**Merge Sort** (Red line) divides a list into two halves, recursively sorts each half, and then merges the sorted halves to produce a sorted list. Merge sort merges the two sorted partitions into a single list by repeatedly selecting the smallest element from either the left or right partition and adding that element to a temporary merged list. Once fully merged, the elements in the temporary merged list are copied back to the original list. Its runtime is O(N log N) like on the graph. It divides the input in half until a list of 1 element is reached, which requires log N partitioning levels. At each level, the algorithm does about N comparisons selecting and copying elements from the left and right partitions, yielding N \* log N comparisons.



2, 2, 4

3, 5, 10

4, 9, 16

5, 14, 24

6, 20, 32

7, 27, 40

8, 35, 48

9, 44, 58

10, 54, 68

11, 65, 78

12, 77, 88

13, 90, 98

14, 104, 108

15, 119, 118

16, 135, 128

17, 152, 140

18, 170, 152

19, 189, 164

20, 209, 176

21, 230, 188

22, 252, 200

23, 275, 212

24, 299, 224

25, 324, 236

26, 350, 248

27, 377, 260

28, 405, 272

29, 434, 284

30, 464, 296

31, 495, 308

32, 527, 320

33, 560, 334

34, 594, 348

35, 629, 362

36, 665, 376

37, 702, 389

38, 740, 403

39, 779, 418

40, 819, 432

41, 860, 445

42, 902, 459

43, 945, 473

44, 989, 486

45, 1034, 499

46, 1080, 514

47, 1127, 527

48, 1175, 541

49, 1224, 555

50, 1274, 570

51, 1325, 582

52, 1377, 597

53, 1430, 612

54, 1484, 624

55, 1539, 639

56, 1595, 653

57, 1652, 666

58, 1710, 682

59, 1769, 695

60, 1829, 710

61, 1890, 723

62, 1952, 736

63, 2015, 751

64, 2079, 765

65, 2144, 780

66, 2210, 797

67, 2277, 813

68, 2345, 828

69, 2414, 844

70, 2484, 859

71, 2555, 875

72, 2627, 889

73, 2700, 906

74, 2774, 922

75, 2849, 938

76, 2925, 955

77, 3002, 968

78, 3080, 985

79, 3159, 1002

80, 3239, 1016

81, 3320, 1031

82, 3402, 1047

83, 3485, 1063

84, 3569, 1079

85, 3654, 1095

86, 3740, 1111

87, 3827, 1127

88, 3915, 1141

89, 4004, 1158

90, 4094, 1173

91, 4185, 1188

92, 4277, 1205

93, 4370, 1220

94, 4464, 1237

95, 4559, 1254

96, 4655, 1266

97, 4752, 1284

98, 4850, 1301

99, 4949, 1317

100, 5049, 1332