Section 5: Hashing & Sorting

0. Sorting Hat

Suppose we sort an array of numbers, but it turns out every element of the array is the same, e.g.,{17, 17, 17, ..., 17}. (So, in hindsight, the sorting is useless.)

a) What is the asymptotic running time of **insertion** sort in this case?

b) What is the asymptotic running time of **selection** sort in this case?

c) What is the asymptotic running time of **merge** sort in this case?

d) What is the asymptotic running time of **quick** sort in this case?

1. Another Sort of Sorting…

Given an array of integers as such: {11, 13, 55, 67, 79, 10, 8, 6, 4, 2}. Please answer the following questions:

a) What is the asymptotic running time of **insertion** sort for this array?

b) What is the asymptotic running time of **selection** sort in this case?

c) What is the asymptotic running time of **merge** sort in this case?

d) What is the asymptotic running time of **quick** sort in this case (assuming that we choose the leftmost element as the pivot each time)?

2. Radix sort

Use Radix Sort to sort the following list of integers in ascending order:
[170, 45, 75, 90, 802, 24, 2, 66]

(a) Bucket Sort on 1's Digit

0	1	2	3	4	5	6	7	8	9

(b) Bucket Sort on 10's Digit

0	1	2	3	4	5	6	7	8	9

(c) Bucket Sort on 100's Digit

0	1	2	3	4	5	6	7	8	9