Chris Swenson

CSC 130

4/26/16

Assignment 4 Analysis

**Heap Algorithm** – (1 <= K <= 1000000)

K = 10

57ms, 41ms, 30ms, 29ms, 35ms, 27ms, 34ms, 41ms, 35ms, 35ms, AVG: 36ms.

K = 100

56ms, 37ms, 30ms, 39ms, 40ms, 43ms, 33ms, 31ms, 31ms, 37ms, AVG: 37ms.

K = 1000

49ms, 38ms, 32ms, 37ms, 40ms, 39ms, 33ms, 34ms, 41ms, 35ms, AVG: 37ms.

K = 10000

85ms, 70ms, 59ms, 52ms, 58ms, 54ms, 78ms, 72ms, 70ms, 57ms, AVG: 65ms.

K = 100000

138ms, 161ms, 144ms, 140ms, 116ms, 178ms, 142ms, 95ms, 169ms, 110ms, AVG: 139ms.

**Quickselect Algorithm** – (1 <= K <= 1000000)

K = 10

72ms, 26ms, 8ms, 34ms, 3ms, 14ms, 28ms, 25ms, 2ms, 17ms, AVG: 22ms.

K = 100

55ms, 14ms, 22ms, 32ms, 25ms, 13ms, 39ms, 11ms, 24ms, 40ms, AVG: 27ms.

K = 1000

18ms, 35ms, 15ms, 24ms, 35ms, 21ms, 24ms, 26ms, 10ms, 15ms, AVG: 22ms.

K = 10000

52ms, 30ms, 23ms, 11ms, 17ms, 3ms, 25ms, 6ms, 21ms, 11ms, AVG: 19ms.

K = 100000

59ms, 40ms, 32ms, 26ms, 44ms, 24ms, 16ms, 37ms, 41ms, 45ms, AVG: 36ms.

**Analysis** –

Heap has a time complexity of O(nlogn) while Quickselect has a time complexity of O(n2), however, as the chart above shows, Quickselect has a better runtime while finding the Kth term, especially for higher values of K. This is because the average time complexity of Quickselect is O(n) while Heap’s average time complexity is still O(nlogn).