Insert

1963 seconds to sort

Size\_10 = 0.299 s

Size\_20 = 2.158 s

Size\_50 = 38.635 s

Size\_100 = 161.027 s

Size\_500 = 4912.689 s

Size\_1000 = 19962.513 s

Quick

2.357 seconds to sort

Size\_10 = 0.297 s

Size\_20 = 2.368 s

Size\_50 = 32.631 s

Size\_100 = 162.824 s

Size\_500 = 4805.184 s

Size\_1000 = 19608.134 s

Merge

10.114 seconds to sort

Size\_10 = 0.291 s

Size\_20 = 2.378 s

Size\_50 = 33.534 s

Size\_100 = 176.374 s

Size\_500 = 4709.947 s

Size\_1000 = 21056.927 s

Heap

4.984 seconds to sort

Size\_10 = 0.305 s

Size\_20 = 3.001 s

Size\_50 = 35.632 s

Size\_100 = 172.711 s

Size\_500 = 4951.376 s

Size\_1000 = 20153.306 s

Hashing

Size\_10 = 0.593 seconds

Size\_20 = 1.737 seconds

Size\_50 = 8.373 seconds

Size\_100 = 32.631 seconds

Size\_500 = 800.347 seconds

Size\_1000 = 3200.537 seconds

Notice that even though insert, quick, merge, and heap sort all have different sorting times, they all take roughly the same amount of time to find all words in the grid. This is because all those algorithms incorporate a binary search to lookup an element in the dictionary. Hashing takes “0” seconds to sort as it never has to sort elements, only copy them into another location in memory. Hashing also doesn’t use a binary search to find elements in the dictionary, but uses a hash lookup to find a