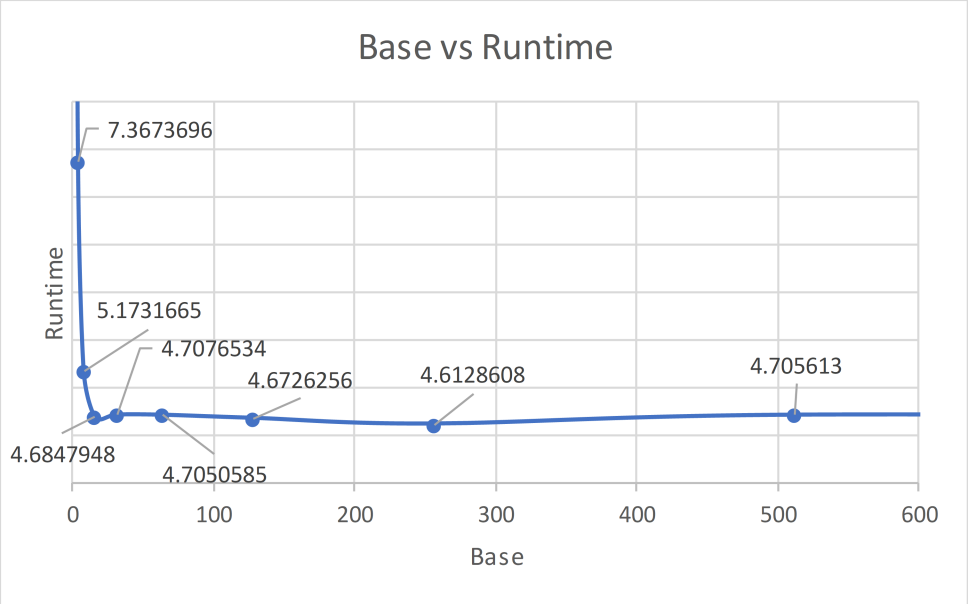
Task 2 PDF

The base I choose for this task is [2, 4, 8, 16, 32 ,64 ,128, 512, 1024], in another word, power of 2. Base 2 will give us maximum number of digit, in which maximize the number of sorts, but the time taken for each sort is the shortest. Increase the base in power of 2 will give me a fast growth rate so that I can compare the condition when base is small to base is quite big.

The following is a graph of base vs runtime of the algorithm.



According to the graph, we can see that the runtime reduced significantly if the base is not 2, this happens because there're only two digits in base 2, so the length of each element will be longest, so the number of stable sort will be maximized. The runtime kept decrease until base 128, then the runtime started to increase. In conclusion, the selection of base is a trade off, if base is low, the number of digit is high, but each sort cost less, on the contrary, if base is high, the number of sort required will be low, but each sort will cost more.