

Task5

Nawar Saeed

Artificial Intelligence for the Web, VT21
DT506A

16 april 2021

What I observe is that the second method takes longer time to run. My observation of why this is so is that in the first method, numbers from $n \rightarrow n - 1$ are added to the list, which may not take that long time compared to the second method where the numbers are shuffled from random positions of the first list. I might be the `random.shuffle()`, in line 39 of Task5.py, that takes some time because of its structure and its time complexity. According to this post¹, so it runs in $O(n)$ so it might be fast enough.

The ten words with the largest number of documents				
N	Result of List1	Result of List2	Running time of List1 in (ms)	Time Running time of List2 in (ms)
10000	49995000	49995000	1.13237497	6.86696398
100000	4999950000	4999950000	12.03576708	80.65888
150000	11249925000	11249925000	18.1757769	124.909916
550000	151249725000	151249725000	65.9389339	519.667274
1000000	499999500000	499999500000	120.0173580	958.762258
1500000	1124999250000	1124999250000	182.006341	1484.520940
3000000	4499998500000	4499998500000	361.636663	3023.596369
3500000	6124998250000	6124998250000	419.9692269	3491.5304260
5000000	12499997500000	12499997500000	603.0614519	5387.613789

¹<https://softwareengineering.stackexchange.com/questions/215737/how-python-random-shuffle-works>