

Program Structures and Algorithms
Spring 2023(SEC 01)

NAME: Shivani Datar
NUID: 002772160

Task:

Your task is to implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. You will consider two different schemes for deciding whether to sort in parallel.

1. A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
2. Recursion depth or the number of available threads. Using this determination, you might decide on an ideal number (t) of separate threads (stick to powers of 2) and arrange for that number of partitions to be parallelized (by preventing recursion after the depth of $\lg t$ is reached).
3. An appropriate combination of these. There is a Main class and the ParSort class in the sort.par package of the INFO6205 repository. The Main class can be used as is but the ParSort class needs to be implemented where you see "TODO..." [it turns out that these TODOs are already implemented]. Unless you have a good reason not to, you should just go along with the Java8-style future implementations provided for you in the class repository. You must prepare a report that shows the results of your experiments and draws a conclusion (or more) about the efficacy of this method of parallelizing sort. Your experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. You should run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile, obviously) and different cutoff schemes.

Output Screenshots:-

Code Changes:-

1. In Main.java :- added new property in class as threadCount which will take the number of threads to be used in the parallel sorting. Created a new object of ForkJoinPool as 'newPool' and passed the desired threadCount in its constructor. The array size is changed to 2000000, 3000000 and 4000000. Setting the ParSort.cutoff respectively 20000, 30000 and 40000 corresponding to the array size in order to get the cutoff ratio(cutoff/array size) until 1. Also changed the System.out.println statement from specifying cutoff and time to just printing time as a convenience form to transfer the time recording to excel sheet.

```

- Shivani Datar
public class Main {
    1 usage
    public static int threadCount = 4;
    2 usages
    public static ForkJoinPool newPool = new ForkJoinPool(threadCount);
    Shivani Datar *
    public static void main(String[] args) {
        processArgs(args);
        //System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        System.out.println("Degree of parallelism: " + newPool.getParallelism());
        Random random = new Random();
        int[] array = new int[3000000];
        //int[] array = new int[1000000];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 0; j < 100; j++) {
            ParSort.cutoff = 30000 * (j + 1);

```

2. In ParSort.java :- Just adding the instance of ForkJoinPool in CompletableFuture method so it can take the number of threads decided.

```

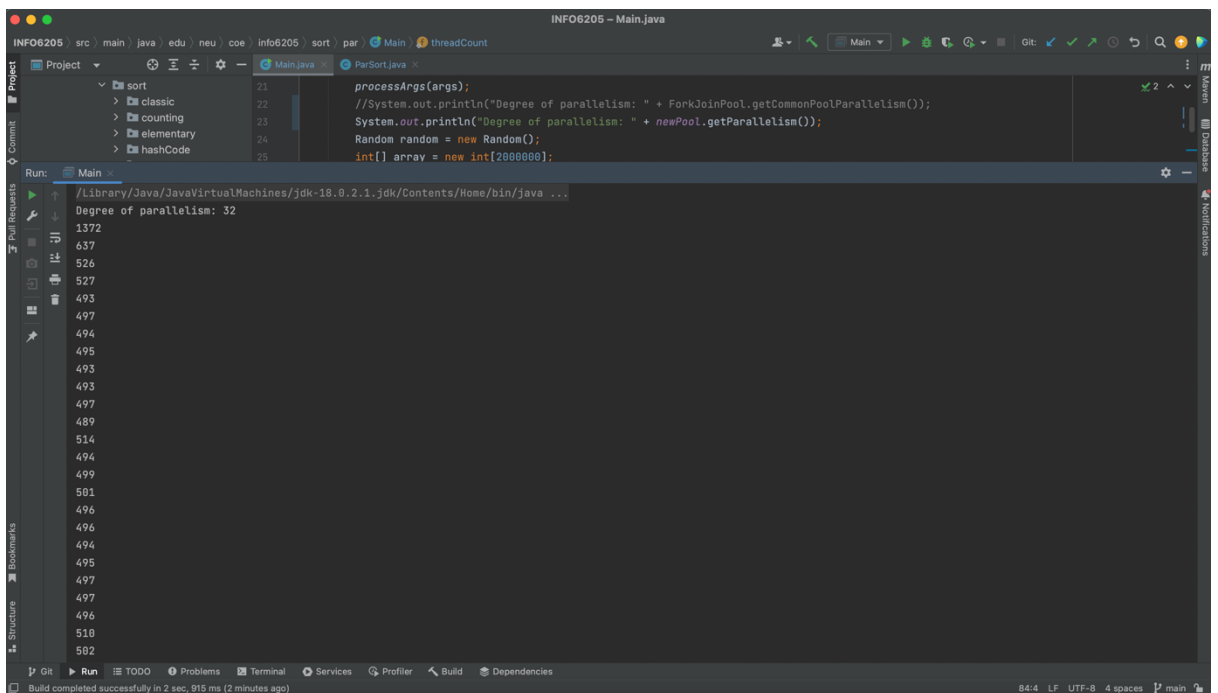
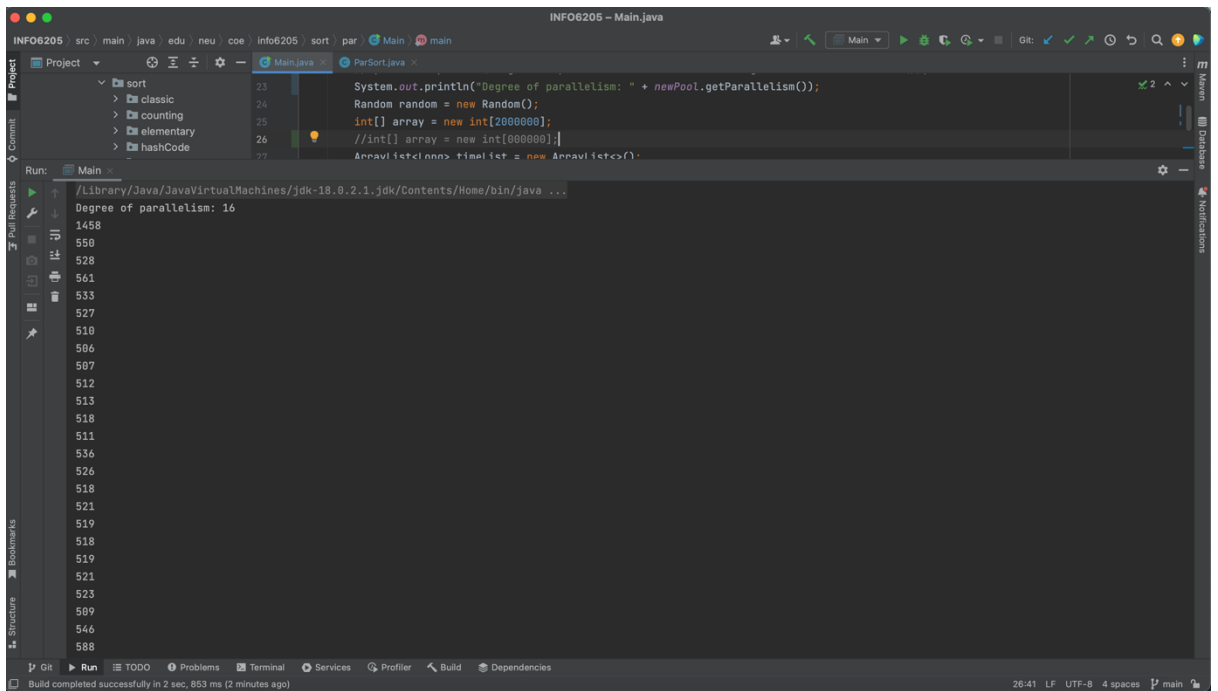
2 usages Shivani Datar *
private static CompletableFuture<int[]> parsort(int[] array, int from, int to) {
    return CompletableFuture.supplyAsync(
        () -> {
            int[] result = new int[to - from];
            // TO IMPLEMENT
            System.arraycopy(array, from, result, destPos: 0, result.length);
            sort(result, from: 0, to: to - from);
            return result;
        }, Main.newPool
    );
}

```

Output Screenshots:- Following are few of the output screenshots:-

```
INFO6205 - Main.java
src \ main \ java \ edu \ neu \ coe \ info6205 \ sort \ par \ Main \ main
Project
  sort
    classic
    counting
    elementary
    hashCode
Run: Main
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1-jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
1248
634
570
598
557
556
645
633
636
637
640
634
733
753
751
721
728
737
737
743
746
731
747
741
732
Build completed successfully in 2 sec, 910 ms (11 minutes ago)
```

```
INFO6205 - Main.java
src \ main \ java \ edu \ neu \ coe \ info6205 \ sort \ par \ Main \ threadCount
Project
  sort
    classic
    counting
    elementary
    hashCode
Run: Main
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1-jdk/Contents/Home/bin/java ...
Degree of parallelism: 8
1308
546
526
603
646
552
520
540
498
498
504
498
521
517
503
511
516
497
503
498
503
488
510
503
505
Build completed successfully in 2 sec, 888 ms (2 minutes ago)
```



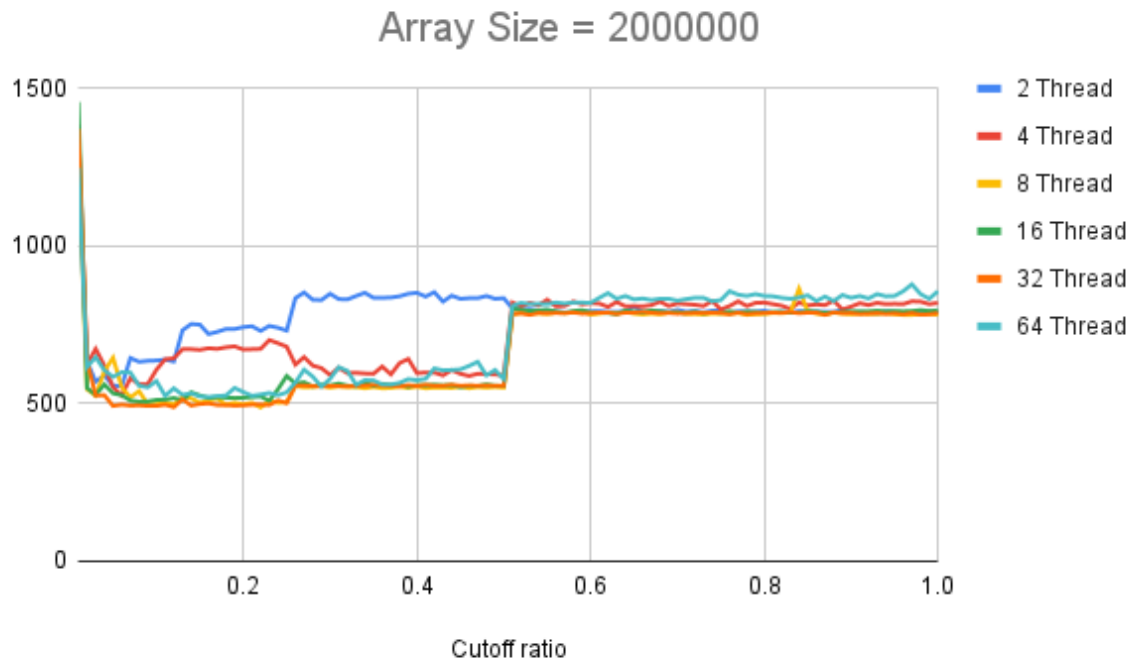
For Array Size = 2000000, for threads 2 to 64. Time is recorded in ms.

Cutoff	Cutoff ratio	2 Thread	4 Thread	8 Thread	16 Thread	32 Thread	64 Thread
20000	0.01	1248	1384	1308	1458	1372	1249
40000	0.02	634	625	546	550	637	616
60000	0.03	570	674	526	528	526	648
80000	0.04	590	621	603	561	527	606
100000	0.05	557	552	646	533	493	585
120000	0.06	556	526	552	527	497	600
140000	0.07	645	581	520	510	494	599
160000	0.08	633	562	540	506	495	557
180000	0.09	636	562	498	507	493	552
200000	0.1	637	608	498	512	493	572
220000	0.11	640	641	504	513	497	525
240000	0.12	634	644	498	518	489	549
260000	0.13	733	673	521	511	514	528
280000	0.14	753	673	517	536	494	533
300000	0.15	751	671	503	526	499	523
320000	0.16	721	676	511	518	501	521
340000	0.17	728	674	516	521	496	524
360000	0.18	737	679	497	519	496	526
380000	0.19	737	682	503	518	494	550
400000	0.2	743	672	498	519	495	537
420000	0.21	746	673	503	521	497	525
440000	0.22	731	674	488	523	497	529
460000	0.23	747	702	510	509	496	534
480000	0.24	741	692	503	546	510	527
500000	0.25	732	681	505	588	502	536
520000	0.26	835	624	556	564	558	567
540000	0.27	853	648	552	568	558	608
560000	0.28	830	621	553	555	556	586
580000	0.29	829	613	557	560	555	554
600000	0.3	849	591	551	558	556	579
620000	0.31	832	614	555	564	556	616
640000	0.32	831	597	553	557	555	604
660000	0.33	842	598	553	555	557	560
680000	0.34	852	596	549	567	558	575
700000	0.35	836	595	553	557	560	574
720000	0.36	836	618	549	563	558	561
740000	0.37	837	593	550	558	558	561
760000	0.38	841	629	554	559	555	566

780000	0.39	849	642	550	554	559	578
800000	0.4	852	597	554	559	556	575
820000	0.41	840	599	549	558	560	579
840000	0.42	854	599	551	560	557	612
860000	0.43	824	590	551	558	557	606
880000	0.44	843	607	550	559	560	607
900000	0.45	833	595	554	553	557	609
920000	0.46	835	588	549	557	558	620
940000	0.47	835	595	553	556	557	633
960000	0.48	842	594	551	561	558	590
980000	0.49	833	593	553	557	557	607
1000000	0.5	835	590	551	556	557	578
1020000	0.51	802	821	784	802	785	811
1040000	0.52	798	808	788	798	787	820
1060000	0.53	800	820	787	795	782	816
1080000	0.54	795	812	787	796	788	817
1100000	0.55	797	830	785	798	787	816
1120000	0.56	791	809	789	793	788	821
1140000	0.57	790	810	788	788	787	818
1160000	0.58	790	824	789	794	790	821
1180000	0.59	793	816	793	797	786	821
1200000	0.6	793	819	783	790	787	820
1220000	0.61	794	818	786	789	789	836
1240000	0.62	791	807	790	788	789	851
1260000	0.63	791	823	786	785	785	833
1280000	0.64	793	809	791	795	789	842
1300000	0.65	808	810	786	792	789	831
1320000	0.66	792	809	783	790	792	834
1340000	0.67	791	813	786	796	789	829
1360000	0.68	794	824	784	793	787	833
1380000	0.69	791	810	789	790	788	833
1400000	0.7	798	818	791	789	786	827
1420000	0.71	792	816	785	788	787	832
1440000	0.72	794	814	785	792	788	836
1460000	0.73	790	810	783	787	788	836
1480000	0.74	795	826	786	790	789	825
1500000	0.75	795	799	786	788	788	829
1520000	0.76	787	810	786	793	784	857
1540000	0.77	793	825	788	789	789	846
1560000	0.78	792	810	782	792	785	842
1580000	0.79	792	820	784	791	787	848
1600000	0.8	795	821	785	788	788	842

1620000	0.81	789	816	788	791	789	841
1640000	0.82	791	811	788	791	788	836
1660000	0.83	789	801	789	789	791	833
1680000	0.84	794	821	865	789	787	837
1700000	0.85	792	812	793	793	789	844
1720000	0.86	792	815	786	792	789	826
1740000	0.87	791	813	784	788	782	840
1760000	0.88	791	827	789	789	790	826
1780000	0.89	797	799	786	791	787	845
1800000	0.9	789	808	784	793	788	836
1820000	0.91	790	819	786	793	785	842
1840000	0.92	789	813	785	793	786	834
1860000	0.93	786	820	785	795	788	848
1880000	0.94	793	816	788	793	785	841
1900000	0.95	791	817	784	794	787	842
1920000	0.96	787	816	782	793	784	857
1940000	0.97	788	825	782	794	788	879
1960000	0.98	789	824	784	797	783	848
1980000	0.99	792	816	785	794	784	833
2000000	1	789	820	782	796	788	857

Graph :-



For Array Size = 3000000, for threads 2 to 64. Time is recorded in ms.

Cutoff	Cutoff ratio	2 Thread	4 Thread	8 Thread	16 Thread	32 Thread	64 Thread
30000	0.01	1728	1678	1739	1583	1704	1610
60000	0.02	1092	921	878	859	877	918
90000	0.03	1035	817	832	796	785	1012
120000	0.04	1086	824	823	876	766	821
150000	0.05	949	780	791	857	743	804
180000	0.06	955	781	800	835	734	783
210000	0.07	1027	823	816	803	754	737
240000	0.08	1055	828	825	821	756	754
270000	0.09	1052	823	809	873	760	753
300000	0.1	1176	822	805	831	759	746
330000	0.11	1028	820	805	839	772	742
360000	0.12	1004	822	802	816	779	742
390000	0.13	1158	882	834	756	771	739
420000	0.14	1150	886	893	775	754	747
450000	0.15	1178	884	942	758	757	738
480000	0.16	1171	885	956	750	757	734
510000	0.17	1142	903	958	734	763	737
540000	0.18	1181	886	946	778	760	725
570000	0.19	1161	898	961	775	773	730
600000	0.2	1147	914	869	768	771	723
630000	0.21	1145	897	863	864	754	741
660000	0.22	1157	890	908	917	761	732
690000	0.23	1156	923	832	852	751	733
720000	0.24	1158	892	786	808	764	739
750000	0.25	1143	921	856	791	772	729
780000	0.26	1278	850	914	907	871	841
810000	0.27	1302	900	934	863	867	845
840000	0.28	1379	844	928	923	871	850
870000	0.29	1292	837	944	928	868	845
900000	0.3	1300	843	946	945	867	850
930000	0.31	1315	848	976	970	867	847
960000	0.32	1312	886	995	981	869	846
990000	0.33	1374	840	939	972	863	846
1020000	0.34	1345	844	884	983	853	851
1050000	0.35	1390	854	892	880	854	845
1080000	0.36	1453	845	874	913	853	847
1110000	0.37	1431	843	900	919	853	849
1140000	0.38	1418	843	951	916	855	845
1170000	0.39	1470	845	954	923	850	849

1200000	0.4	1412	843	956	939	857	847
1230000	0.41	1455	843	953	967	848	844
1260000	0.42	1537	865	952	1005	851	846
1290000	0.43	1396	839	974	876	851	850
1320000	0.44	1412	841	1002	916	856	844
1350000	0.45	1402	840	939	908	844	851
1380000	0.46	1489	839	933	875	855	849
1410000	0.47	1495	843	987	925	852	846
1440000	0.48	1509	842	1042	922	855	849
1470000	0.49	1544	842	1033	927	859	846
1500000	0.5	1575	837	1055	924	859	845
1530000	0.51	1296	1204	1313	1258	1216	1228
1560000	0.52	1286	1195	1279	1268	1211	1220
1590000	0.53	1327	1202	1284	1277	1214	1218
1620000	0.54	1307	1198	1296	1276	1208	1226
1650000	0.55	1292	1191	1366	1277	1210	1216
1680000	0.56	1309	1195	1291	1287	1208	1217
1710000	0.57	1309	1204	1301	1285	1211	1220
1740000	0.58	1283	1194	1293	1268	1214	1223
1770000	0.59	1294	1198	1292	1261	1210	1218
1800000	0.6	1335	1204	1298	1259	1208	1212
1830000	0.61	1328	1198	1295	1259	1207	1224
1860000	0.62	1304	1202	1291	1271	1210	1217
1890000	0.63	1319	1198	1304	1280	1214	1222
1920000	0.64	1305	1203	1291	1275	1206	1218
1950000	0.65	1307	1198	1300	1260	1207	1217
1980000	0.66	1305	1201	1303	1274	1217	1216
2010000	0.67	1328	1203	1336	1268	1219	1221
2040000	0.68	1300	1201	1294	1289	1208	1223
2070000	0.69	1340	1199	1313	1276	1213	1219
2100000	0.7	1304	1203	1308	1283	1204	1216
2130000	0.71	1291	1199	1300	1282	1205	1219
2160000	0.72	1313	1202	1322	1278	1223	1216
2190000	0.73	1302	1204	1335	1309	1220	1214
2220000	0.74	1316	1209	1282	1253	1212	1222
2250000	0.75	1328	1200	1294	1273	1206	1215
2280000	0.76	1349	1204	1314	1280	1211	1216
2310000	0.77	1319	1198	1299	1267	1222	1221
2340000	0.78	1323	1200	1307	1277	1207	1215
2370000	0.79	1314	1202	1309	1268	1209	1212
2400000	0.8	1355	1202	1316	1247	1204	1218
2430000	0.81	1319	1208	1342	1280	1213	1220

2460000	0.82	1317	1202	1331	1273	1203	1219
2490000	0.83	1360	1204	1319	1307	1211	1212
2520000	0.84	1307	1201	1347	1296	1210	1220
2550000	0.85	1281	1205	1344	1288	1208	1215
2580000	0.86	1279	1202	1321	1283	1211	1227
2610000	0.87	1279	1207	1347	1291	1212	1221
2640000	0.88	1280	1204	1334	1286	1203	1220
2670000	0.89	1278	1200	1360	1287	1200	1217
2700000	0.9	1285	1202	1280	1342	1204	1212
2730000	0.91	1283	1199	1284	1266	1204	1221
2760000	0.92	1284	1198	1289	1251	1202	1224
2790000	0.93	1283	1205	1284	1250	1209	1217
2820000	0.94	1286	1200	1300	1326	1206	1219
2850000	0.95	1284	1202	1288	1263	1207	1216
2880000	0.96	1291	1195	1304	1257	1202	1217
2910000	0.97	1296	1192	1299	1263	1208	1219
2940000	0.98	1304	1197	1296	1260	1203	1213
2970000	0.99	1312	1198	1303	1255	1206	1214
3000000	1	1333	1196	1304	1258	1213	1218

Graph:-



For Array Size = 4000000, for threads 2 to 64. Time is recorded in ms.

Cutoff	Cutoff ratio	2 Thread	4 Thread	8 Thread	16 Thread	32 Thread	64 Thread
40000	0.01	2558	2269	1923	1919	1862	1894
80000	0.02	1550	1425	1208	1096	1074	1113
120000	0.03	1612	1432	1135	1101	1067	1102
160000	0.04	1637	1385	1088	1069	1077	1113
200000	0.05	1586	1401	1042	1041	1062	1023
240000	0.06	1524	1409	1110	1021	1060	1037
280000	0.07	1739	1573	1061	1012	1102	1019
320000	0.08	1765	1431	1065	1032	1019	1025
360000	0.09	1768	1429	1073	1022	987	1017
400000	0.1	1749	1448	1064	1040	986	1003
440000	0.11	1744	1431	1077	1047	1007	1015
480000	0.12	1777	1440	1067	1000	1069	1020
520000	0.13	2086	1658	1134	989	969	989
560000	0.14	2213	1553	1199	997	963	986
600000	0.15	2266	1550	1269	1007	970	1009
640000	0.16	2202	1545	1259	987	958	1017
680000	0.17	2325	1551	1184	985	978	990
720000	0.18	2313	1552	1141	968	1019	976
760000	0.19	2117	1547	1138	968	965	996
800000	0.2	1968	1537	1096	973	975	983
840000	0.21	1925	1541	1085	965	963	1003
880000	0.22	2001	1535	1017	975	966	977
920000	0.23	1960	1533	969	975	959	978
960000	0.24	1973	1545	953	958	971	974
1000000	0.25	2039	1540	972	967	973	1001
1040000	0.26	2235	1470	1155	1139	1142	1153
1080000	0.27	2287	1464	1160	1144	1145	1152
1120000	0.28	2199	1459	1144	1134	1141	1147
1160000	0.29	2254	1454	1151	1143	1150	1149
1200000	0.3	2203	1463	1143	1138	1142	1150
1240000	0.31	2248	1458	1147	1141	1151	1153
1280000	0.32	2249	1468	1147	1151	1142	1149
1320000	0.33	2229	1484	1150	1138	1145	1153
1360000	0.34	2252	1463	1148	1148	1143	1151
1400000	0.35	2193	1468	1148	1146	1147	1151
1440000	0.36	2246	1468	1150	1136	1147	1148
1480000	0.37	2240	1467	1151	1146	1146	1146
1520000	0.38	2255	1460	1150	1159	1147	1150
1560000	0.39	2269	1473	1152	1149	1143	1145
1600000	0.4	2259	1471	1147	1143	1143	1148

1640000	0.41	2278	1477	1157	1138	1144	1150
1680000	0.42	2448	1478	1151	1151	1148	1148
1720000	0.43	2380	1468	1217	1145	1146	1147
1760000	0.44	2366	1468	1243	1140	1154	1151
1800000	0.45	2353	1479	1251	1141	1143	1150
1840000	0.46	2375	1465	1251	1139	1150	1147
1880000	0.47	2407	1461	1252	1152	1139	1153
1920000	0.48	2264	1462	1257	1147	1151	1152
1960000	0.49	2390	1459	1267	1145	1144	1150
2000000	0.5	2411	1470	1164	1171	1148	1151
2040000	0.51	2265	2146	1697	1650	1649	1662
2080000	0.52	2524	2151	1707	1655	1646	1653
2120000	0.53	2266	2152	1681	1645	1651	1668
2160000	0.54	2244	2148	1696	1647	1652	1659
2200000	0.55	2301	2142	1758	1645	1652	1657
2240000	0.56	2306	2144	1759	1637	1648	1656
2280000	0.57	2267	2147	1778	1638	1655	1651
2320000	0.58	2192	2142	1756	1647	1659	1659
2360000	0.59	2203	2148	1720	1647	1646	1663
2400000	0.6	2203	2150	1748	1648	1653	1668
2440000	0.61	2207	2152	1709	1651	1654	1666
2480000	0.62	2213	2144	1708	1640	1650	1652
2520000	0.63	2194	2138	1715	1643	1644	1661
2560000	0.64	2202	2146	1710	1655	1646	1661
2600000	0.65	2199	2159	1704	1646	1647	1664
2640000	0.66	2232	2158	1709	1640	1655	1657
2680000	0.67	2264	2157	1712	1640	1656	1664
2720000	0.68	2243	2156	1715	1651	1645	1663
2760000	0.69	2231	2160	1732	1663	1649	1659
2800000	0.7	2246	2149	1723	1644	1649	1660
2840000	0.71	2284	2155	1741	1651	1648	1651
2880000	0.72	2279	2157	1739	1645	1650	1656
2920000	0.73	2235	2141	1712	1636	1652	1658
2960000	0.74	2231	2137	1658	1638	1645	1656
3000000	0.75	2216	2160	1655	1654	1647	1655
3040000	0.76	2304	2150	1657	1654	1648	1660
3080000	0.77	2307	2158	1672	1641	1649	1654
3120000	0.78	2305	2144	1706	1645	1649	1654
3160000	0.79	2274	2145	1759	1657	1646	1660
3200000	0.8	2327	2144	1804	1672	1647	1658
3240000	0.81	2268	2142	1760	1655	1650	1652
3280000	0.82	2262	2145	1761	1649	1645	1663

3320000	0.83	2190	2144	1770	1642	1654	1652
3360000	0.84	2183	2143	1711	1654	1653	1664
3400000	0.85	2176	2140	1722	1680	1645	1655
3440000	0.86	2194	2152	1705	1652	1645	1653
3480000	0.87	2204	2145	1703	1678	1667	1655
3520000	0.88	2201	2144	1704	1649	1652	1655
3560000	0.89	2195	2147	1701	1643	1648	1661
3600000	0.9	2203	2151	1705	1642	1645	1656
3640000	0.91	2201	2147	1710	1643	1643	1656
3680000	0.92	2241	2168	1716	1661	1650	1664
3720000	0.93	2261	2154	1710	1645	1665	1654
3760000	0.94	2243	2154	1722	1641	1651	1656
3800000	0.95	2150	2150	1709	1642	1651	1649
3840000	0.96	2163	2146	1734	1638	1655	1651
3880000	0.97	2190	2150	1752	1655	1651	1657
3920000	0.98	2336	2147	1746	1645	1648	1655
3960000	0.99	2220	2149	1648	1635	1646	1657
4000000	1	2216	2149	1652	1645	1654	1652

Graph :-

