Program Structures and Algorithms
Spring 2023(SEC –01)

NAME: Harshini Venkata Chalam
NUID: 002934047
Assignment 5

Task:

To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. Considering two different schemes for deciding whether to sort in parallel.

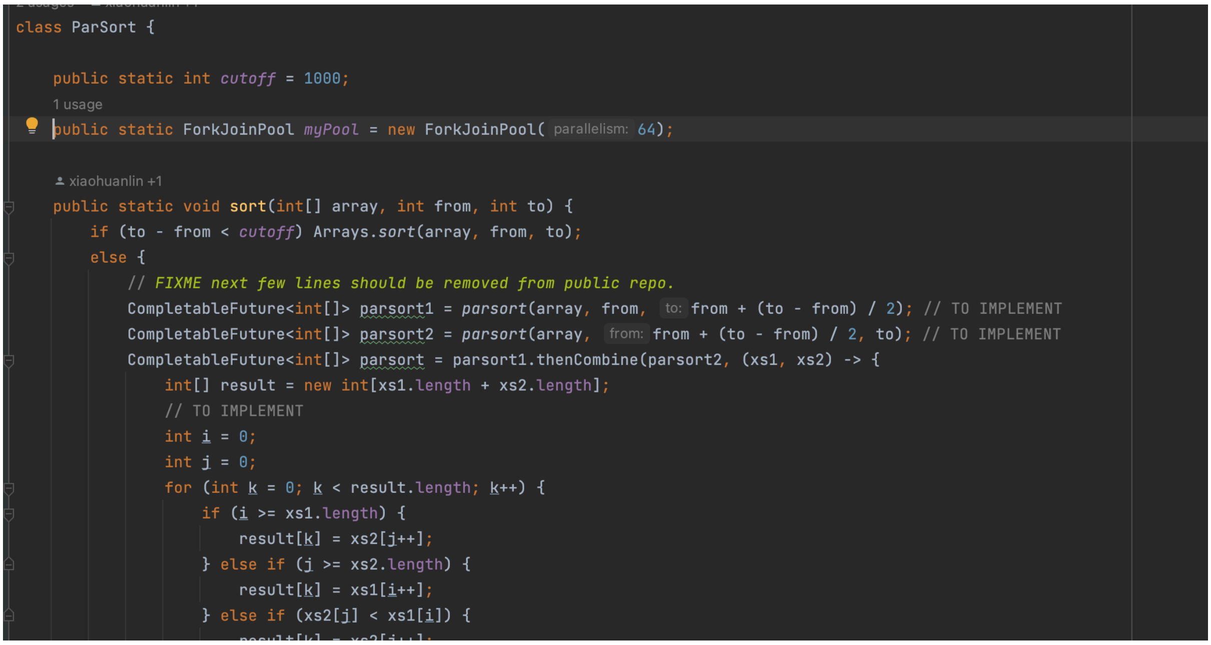
1. A cutoff (defaults to, say, 1000) which is updated according to the first argument in the command line when running. To experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
2. Recursion depth or 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.

Relationship Conclusion:

As the array size increases the time taken to sort also increases. Parallel sort performance does not vary much as the thread count increases. When the array size is very small, the overhead of creating and synchronizing threads may be larger than the time it takes to sort the array sequentially. As the size of the array increases, the benefits of parallelism become more evident, as parallel sorting allows multiple threads to work instantaneously to sort the array. As per the below values and observations for various thread counts, increasing the number of threads can improve performance of parallel sorting up to a certain point, after that advantages of extra threads may be reduced .

It also depends the size of the array, characteristics of the system being used as well.Also it can be observed that the performance is improved while cut off is below 50% of the given array size.
As the cut off increases or is equal to 50% of the array size performance degrades or remains almost constant.

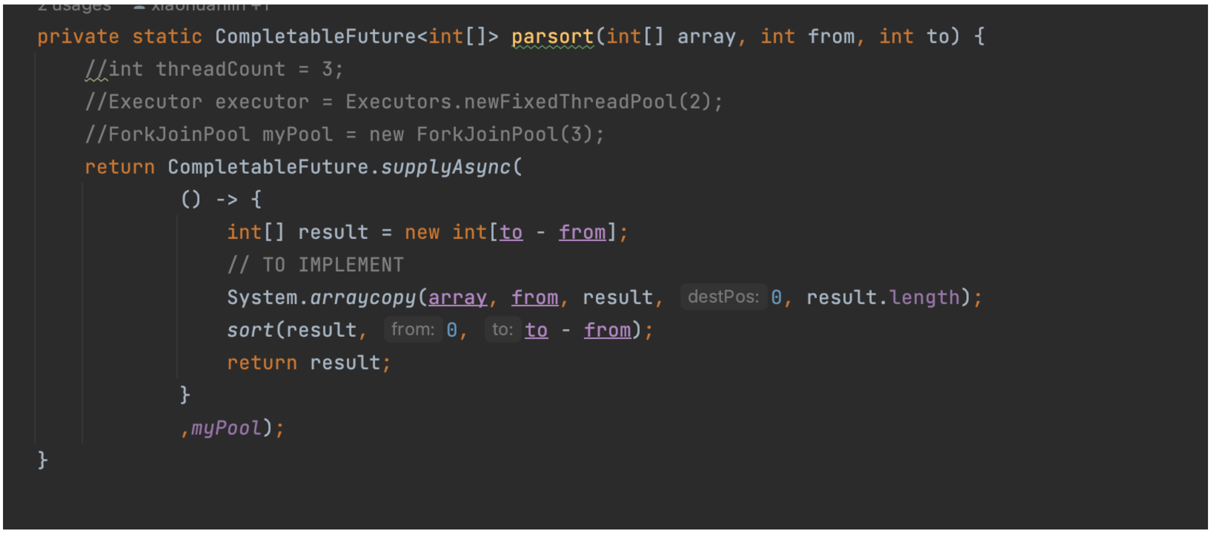
Evidence to support that conclusion:



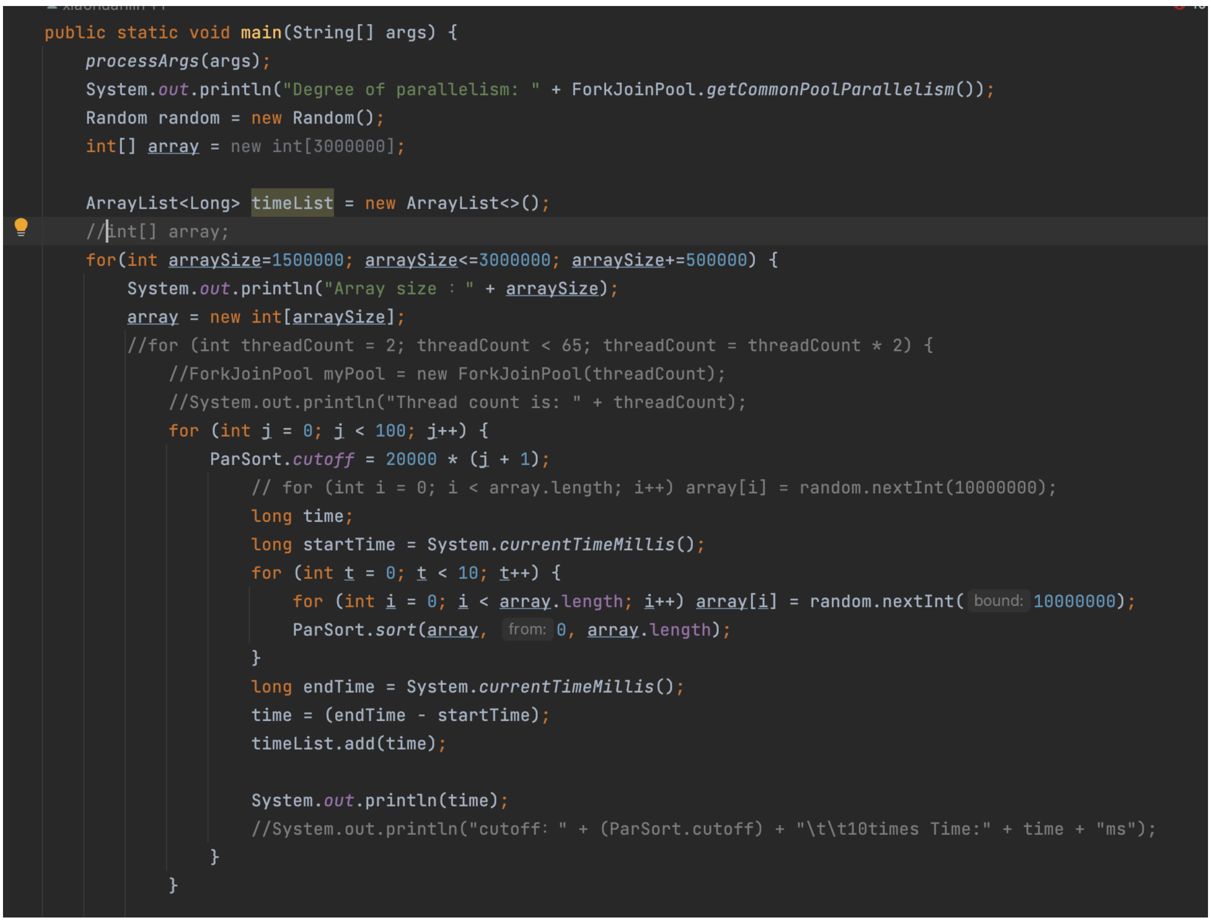
```
2 usages  xiaohuanlin +1
class ParSort {

    public static int cutoff = 1000;
    1 usage
    public static ForkJoinPool myPool = new ForkJoinPool( parallelism: 64);

    xiaohuanlin +1
    public static void sort(int[] array, int from, int to) {
        if (to - from < cutoff) Arrays.sort(array, from, to);
        else {
            // FIXME next few lines should be removed from public repo.
            CompletableFuture<int[]> parsort1 = parsort(array, from,  to: from + (to - from) / 2); // TO IMPLEMENT
            CompletableFuture<int[]> parsort2 = parsort(array,  from: from + (to - from) / 2, to); // TO IMPLEMENT
            CompletableFuture<int[]> parsort = parsort1.thenCombine(parsort2, (xs1, xs2) -> {
                int[] result = new int[xs1.length + xs2.length];
                // TO IMPLEMENT
                int i = 0;
                int j = 0;
                for (int k = 0; k < result.length; k++) {
                    if (i >= xs1.length) {
                        result[k] = xs2[j++];
                    } else if (j >= xs2.length) {
                        result[k] = xs1[i++];
                    } else if (xs2[j] < xs1[i]) {
                        result[k] = xs2[j++];
```



```
2 usages  xiaohuanlin +1
private static CompletableFuture<int[]> parsort(int[] array, int from, int to) {
    //int threadCount = 3;
    //Executor executor = Executors.newFixedThreadPool(2);
    //ForkJoinPool myPool = new ForkJoinPool(3);
    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;
            }
            ,myPool);
}
```



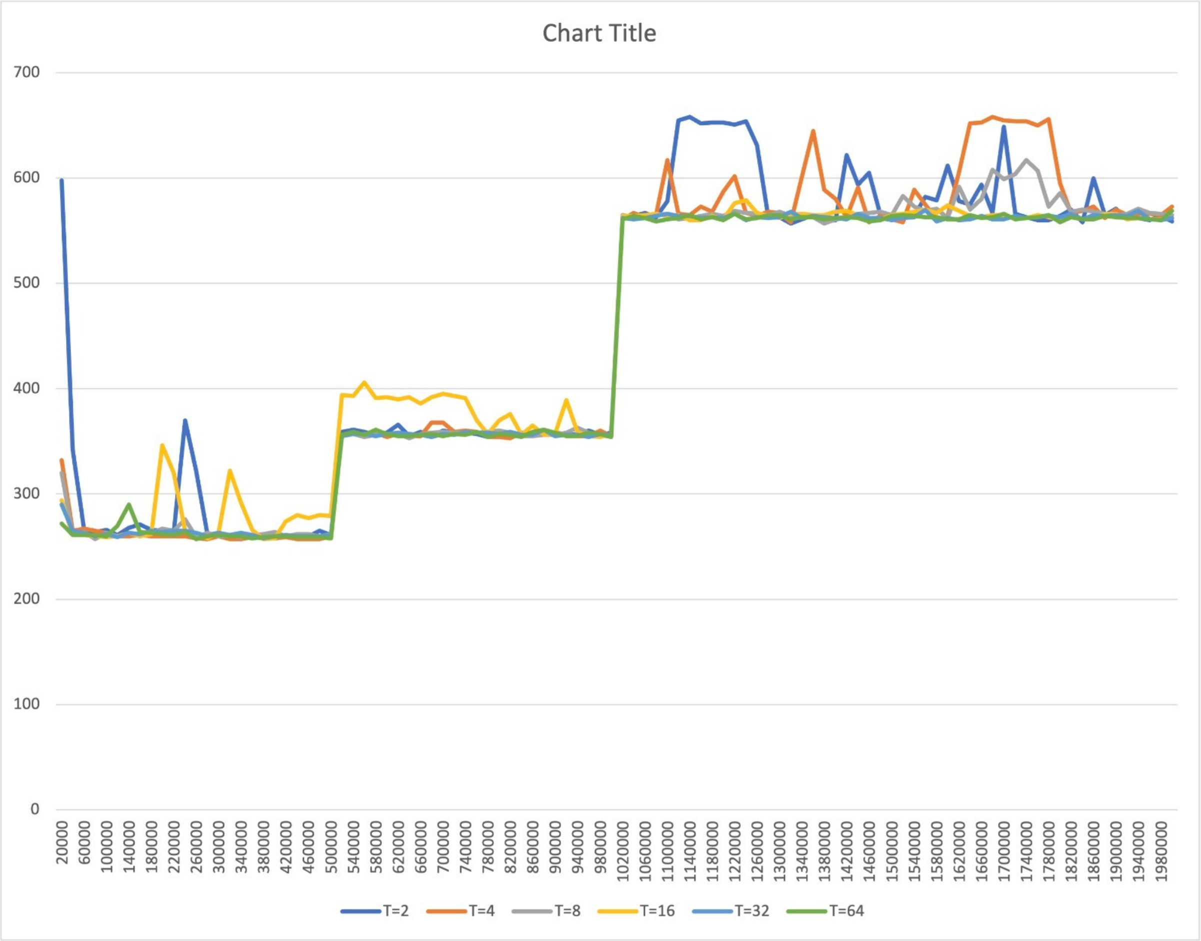
```
public static void main(String[] args) {
    processArgs(args);
    System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
    Random random = new Random();
    int[] array = new int[3000000];

    ArrayList<Long> timeList = new ArrayList<>();
    //int[] array;
    for(int arraySize=1500000; arraySize<=3000000; arraySize+=500000) {
        System.out.println("Array size : " + arraySize);
        array = new int[arraySize];
        //for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2) {
            //ForkJoinPool myPool = new ForkJoinPool(threadCount);
            //System.out.println("Thread count is: " + threadCount);
            for (int j = 0; j < 100; j++) {
                ParSort.cutoff = 20000 * (j + 1);
                    // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
                    long time;
                    long startTime = System.currentTimeMillis();
                    for (int t = 0; t < 10; t++) {
                        for (int i = 0; i < array.length; i++) array[i] = random.nextInt( bound: 10000000);
                        ParSort.sort(array,  from: 0, array.length);
                    }
                    long endTime = System.currentTimeMillis();
                    time = (endTime - startTime);
                    timeList.add(time);

                    System.out.println(time);
                    //System.out.println("cutoff: " + (ParSort.cutoff) + "\t\t10times Time:" + time + "ms");
                }
            }
```

Graphical Representation:

Array Size : 1500000



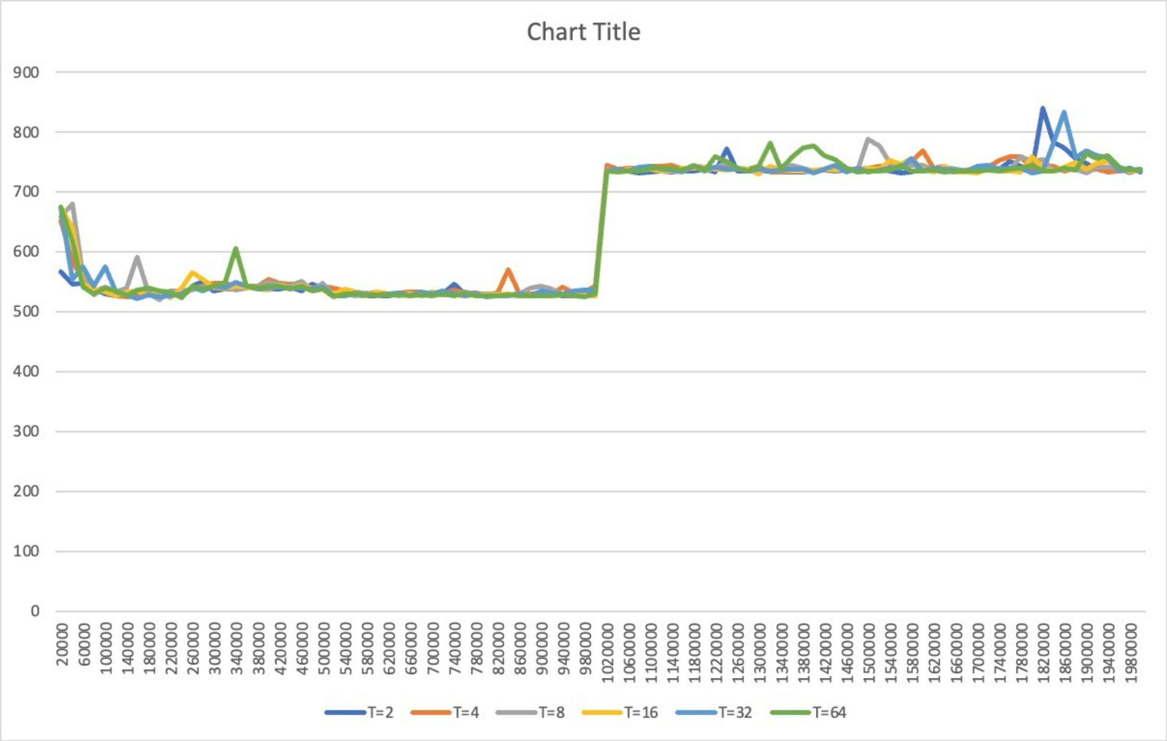
cutoff	T=2(ms)	T=4(ms)	T=8(ms)	T=16(ms)	T=32(ms)	T=64(ms)
20000	1773	792	826	821	1283	849
40000	785	650	675	657	701	656
60000	693	604	589	647	602	616
80000	604	607	604	748	618	595
100000	613	566	579	705	565	573
120000	584	555	560	744	564	619
140000	570	568	551	678	584	554
160000	557	579	562	689	598	644
180000	548	570	589	623	603	712
200000	590	607	590	549	538	621
220000	584	612	643	586	606	599
240000	605	615	629	639	581	578
260000	592	631	631	613	583	608

280000	624	707	770	619	620	638
300000	600	728	945	702	581	653
320000	605	801	636	573	561	629
340000	621	788	623	582	639	682
360000	611	779	680	598	659	682
380000	669	915	801	632	713	674
400000	626	585	635	868	551	562
420000	600	579	640	658	578	556
440000	727	609	610	602	558	582
460000	581	598	647	605	565	562
480000	574	699	635	646	622	603
500000	565	754	668	650	590	603
520000	565	714	628	646	571	613
540000	584	949	752	658	613	616
560000	611	1247	680	811	619	648
580000	626	1399	596	862	583	623
600000	1038	1321	568	1037	576	635
620000	1323	1763	601	917	601	629
640000	635	1035	664	1012	596	688
660000	687	771	586	809	717	1080
680000	591	1041	727	700	875	791
700000	638	707	558	624	582	703
720000	751	667	567	646	557	564
740000	723	711	577	632	561	593
760000	860	822	705	811	697	818
780000	691	735	744	715	709	837
800000	668	689	847	713	691	738
820000	889	683	861	733	680	694
840000	709	776	718	738	692	727
860000	725	737	728	688	703	726
880000	755	685	739	724	691	703
900000	733	678	731	748	675	708
920000	711	786	724	994	703	716
940000	707	808	750	842	730	722
960000	723	892	719	736	681	747
980000	718	693	708	882	694	720
1000000	707	712	716	915	695	704
1020000	688	730	743	732	681	719
1040000	720	699	806	712	724	707
1060000	731	703	775	680	701	1068
1080000	813	716	1117	719	737	741
1100000	764	1090	1151	691	734	755

1120000	753	769	827	705	727	693
1140000	742	771	932	677	833	848
1160000	729	800	874	677	741	873
1180000	701	704	868	695	700	779
1200000	706	725	768	688	674	763
1220000	732	718	683	658	681	741
1240000	698	687	767	673	678	727
1260000	731	706	741	671	717	741
1280000	722	815	846	667	808	939
1300000	710	683	665	685	663	706
1320000	739	688	676	695	657	706
1340000	749	679	708	755	656	708
1360000	745	684	689	835	660	702
1380000	772	688	755	831	673	742
1400000	772	679	743	783	679	710
1420000	746	673	694	702	675	712
1440000	894	672	721	703	747	863
1460000	784	689	724	691	723	742
1480000	1177	692	740	682	713	732
1500000	739	694	671	678	689	1104
1520000	1149	1125	1139	1115	1133	1212
1540000	1197	1133	1134	1117	1142	1129
1560000	1152	1136	1160	1113	1138	1190
1580000	1117	1118	1132	1105	1138	1240
1600000	1117	1114	1137	1222	1146	1149
1620000	1122	1142	1224	1182	1138	1153
1640000	1120	1115	1186	1134	1134	1154
1660000	1203	1129	1694	1116	1118	1174
1680000	1208	1125	1628	1129	1124	1156
1700000	1160	1124	1347	1219	1126	1146
1720000	1159	1136	1215	1142	1130	1143
1740000	1129	1131	1182	1181	1128	1220
1760000	1135	1162	1161	1203	1123	1189
1780000	1162	1162	1158	1135	1130	1206
1800000	1133	1146	1180	1134	1129	1179
1820000	1227	1124	1174	1142	1152	1343
1840000	1159	1144	1162	1127	1132	1157
1860000	1258	1420	1133	1137	1125	1177
1880000	1153	1191	1143	1236	1120	1166
1900000	1152	1240	1140	1167	1113	1145
1920000	1137	1172	1136	1152	1114	1145
1940000	1138	1169	1142	1233	1113	1140

1960000	1143	1175	1163	1150	1120	1123
1980000	1132	1159	1137	1150	1123	1144
2000000	1179	1156	1134	1135	1121	1147

Array Size – 2000000



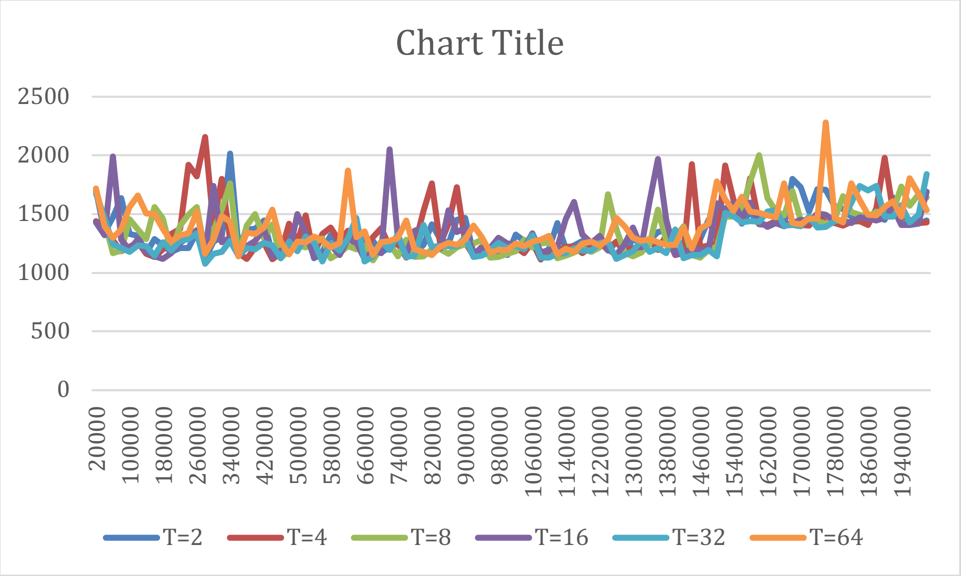
cutoff	T=2	T=4	T=8	T=16	T=32	T=64
20000	1301	1351	1289	1246	1482	1749
40000	634	685	740	634	1015	670
60000	728	600	609	783	614	716
80000	660	627	610	659	607	603
100000	613	628	814	551	619	601
120000	619	582	769	546	726	567
140000	683	600	543	523	593	542
160000	689	596	538	554	565	527
180000	684	607	534	525	605	559
200000	684	631	546	525	631	543
220000	674	600	542	521	583	551
240000	664	613	535	515	582	549
260000	851	667	594	529	587	582
280000	880	672	669	578	599	580

300000	892	681	612	563	628	617
320000	877	672	642	568	595	629
340000	854	691	660	602	624	635
360000	881	745	649	610	656	645
380000	933	712	652	666	609	631
400000	928	806	638	525	633	661
420000	912	624	665	534	665	654
440000	799	645	838	545	644	1084
460000	836	647	578	601	555	537
480000	855	669	522	570	570	551
500000	862	706	618	585	573	574
520000	964	620	625	669	665	629
540000	1009	629	630	591	643	618
560000	1056	719	651	598	652	637
580000	1094	600	602	600	705	661
600000	940	605	675	577	626	691
620000	935	606	599	557	617	603
640000	941	607	606	563	625	620
660000	937	635	637	562	616	600
680000	954	661	633	601	631	620
700000	1009	622	651	647	652	633
720000	985	639	656	655	655	636
740000	958	614	639	713	665	639
760000	1053	667	668	579	665	624
780000	995	630	657	593	629	654
800000	966	669	624	599	659	646
820000	1037	687	685	610	660	659
840000	923	684	669	600	745	710
860000	921	603	731	612	719	744
880000	928	606	602	671	612	609
900000	941	607	607	856	627	604
920000	935	614	609	696	648	635
940000	942	628	625	649	654	619
960000	928	632	620	644	665	613
980000	993	623	642	651	718	674
1000000	932	609	642	717	657	665
1020000	872	864	889	853	893	919
1040000	889	881	902	858	926	886
1060000	919	875	887	867	898	929
1080000	914	908	967	858	915	897
1100000	938	889	946	858	887	904
1120000	991	875	977	859	888	894

1140000	872	889	1007	862	892	912
1160000	870	989	903	866	900	910
1180000	869	866	880	872	902	920
1200000	871	859	991	865	1012	998
1220000	869	871	870	858	881	872
1240000	872	865	870	897	878	882
1260000	870	867	870	874	878	875
1280000	872	875	874	862	877	872
1300000	871	869	878	861	877	882
1320000	878	873	877	959	947	888
1340000	870	873	877	899	918	892
1360000	899	871	875	869	878	891
1380000	873	870	883	886	911	879
1400000	867	870	878	880	886	873
1420000	897	913	874	878	879	878
1440000	870	944	900	870	879	905
1460000	891	897	879	888	922	903
1480000	963	895	890	1004	893	908
1500000	936	915	913	869	892	897
1520000	903	906	880	878	899	1003
1540000	908	893	902	857	896	899
1560000	907	972	881	858	894	886
1580000	934	903	894	860	890	886
1600000	899	902	886	857	890	889
1620000	1017	894	897	857	900	891
1640000	886	900	882	854	895	974
1660000	885	1003	877	858	898	958
1680000	883	869	879	857	888	909
1700000	882	873	1001	862	980	1029
1720000	882	888	872	865	877	868
1740000	879	885	868	908	874	893
1760000	875	880	869	862	874	879
1780000	890	867	885	868	874	874
1800000	871	877	868	864	875	876
1820000	876	872	878	877	875	872
1840000	885	871	879	874	875	869
1860000	907	871	876	875	877	870
1880000	883	877	878	875	885	877
1900000	895	875	876	860	881	871
1920000	882	900	878	865	889	874
1940000	906	885	892	863	898	914
1960000	911	892	925	879	905	897

1980000	901	898	882	1036	905	871
2000000	892	875	893	855	890	883

Array Size – 2500000

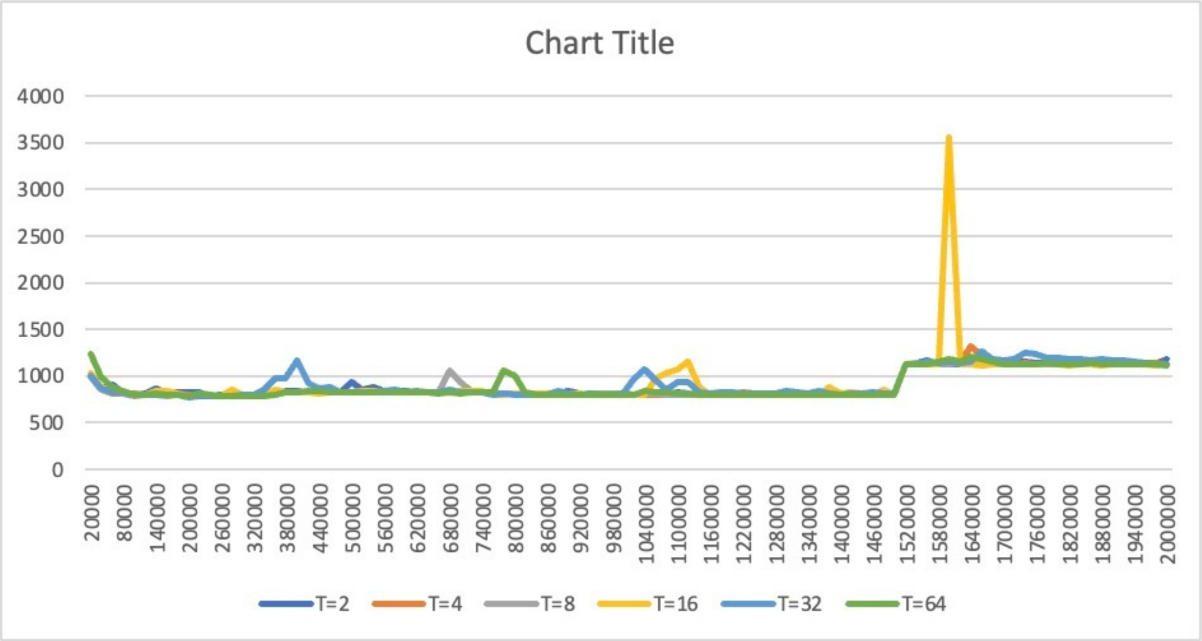


cutoff	T=2	T=4	T=8	T=16	T=32	T=64
20000	1392	1356	1350	1223	1171	1352
40000	1038	1065	1204	948	1047	1183
60000	1106	1086	1388	988	1182	1114
80000	1111	1715	1208	1144	1037	1558
100000	1334	1319	1241	1118	1360	1326
120000	1342	1290	1183	1150	1328	1143
140000	1024	1150	1177	1189	1332	1416
160000	951	948	951	917	1019	1205
180000	1111	1319	1082	1027	1136	1095
200000	957	1037	938	1059	894	1091
220000	1259	1137	1127	1172	1026	1199
240000	1730	891	1256	960	1076	912
260000	1437	981	1176	1012	1169	1057
280000	1036	948	934	969	934	1053
300000	1446	1166	993	1091	1031	1139
320000	1055	933	1019	969	1050	977
340000	1060	963	1027	961	1049	1594
360000	1149	1016	1110	977	1085	1132
380000	940	1024	968	1069	924	1035
400000	1001	1086	1050	1205	1008	1069
420000	1053	1129	1409	924	1089	1116

440000	1076	1184	1042	910	975	970
460000	1208	1176	1054	973	1091	1044
480000	946	1100	970	1009	1027	1043
500000	984	1075	1026	995	1084	1114
520000	1086	1135	1039	1069	1031	1136
540000	1037	944	1038	897	1015	929
560000	1040	929	1000	935	1043	1164
580000	1799	981	1151	958	1165	1037
600000	1133	1006	909	961	918	999
620000	1038	1001	935	1007	954	1048
640000	1114	1263	995	1081	990	1113
660000	1072	1127	958	953	976	965
680000	1027	966	975	927	975	949
700000	1006	974	974	956	1048	955
720000	1042	963	1094	977	1042	1005
740000	1170	1023	1152	1012	1177	1009
760000	955	954	920	1064	1529	1023
780000	980	984	943	996	974	1335
800000	1048	996	1014	1029	1011	1157
820000	987	1021	968	1171	1017	1058
840000	1075	1428	1010	938	1149	1105
860000	1035	964	997	961	1016	963
880000	1056	981	1029	953	1037	960
900000	1090	984	1005	1024	1052	990
920000	1229	998	1109	1055	1007	1048
940000	986	1263	923	1054	1052	1040
960000	942	1217	932	1016	955	1030
980000	993	1193	1005	994	981	1198
1000000	975	1080	1050	1001	1008	1059
1020000	1132	1049	1050	1129	1075	1093
1040000	1073	1125	1063	971	975	945
1060000	1038	959	1001	996	966	950
1080000	1027	978	1341	1001	989	964
1100000	1084	1087	1036	1009	984	980
1120000	1134	1120	1065	1049	1083	994
1140000	1074	1014	964	1003	949	1024
1160000	971	1006	937	1016	974	1039
1180000	982	1059	1061	1045	982	990
1200000	982	1008	980	1190	999	1085
1220000	1035	1192	994	962	1004	948
1240000	994	951	1008	985	1054	953
1260000	1191	1171	1236	1241	1197	1186

1280000	1197	1216	1488	1186	1193	1185
1300000	1261	1416	1238	1241	1225	1201
1320000	1240	1261	1339	1234	1381	1186
1340000	1279	1289	1209	1258	1334	1494
1360000	1457	1463	1337	1208	1213	1415
1380000	1180	1545	1157	1290	1345	1227
1400000	1192	1367	1155	1234	1168	1192
1420000	1485	1581	1175	1204	1173	1162
1440000	1178	1347	1196	1251	1175	1229
1460000	1192	1319	1188	1214	1190	1235
1480000	1408	1187	1394	1298	1209	1190
1500000	1205	1206	1291	1286	1239	1676
1520000	1374	1229	1254	1372	1310	1294
1540000	1459	1186	1199	1305	1447	1934
1560000	1324	1304	1254	1167	1291	1972
1580000	1254	1392	1234	1138	1243	1488
1600000	1262	1170	1471	1148	1184	1323
1620000	1719	1195	1296	1151	1195	1233
1640000	1287	1146	1250	1143	1218	1357
1660000	1237	1441	1193	1248	1222	1508
1680000	1486	1207	1183	1264	1198	1224
1700000	1329	1546	1237	1308	1240	1238
1720000	1359	1226	1209	1188	1247	1214
1740000	1214	1190	1177	1254	1214	1235
1760000	1354	1485	1352	1195	1422	1400
1780000	1180	1271	1141	1198	1320	1184
1800000	1184	1193	1172	1431	1196	1177
1820000	1183	1185	1206	1486	1171	1269
1840000	1164	1433	1161	1214	1187	1362
1860000	1381	1241	1188	1189	1211	1233
1880000	1447	1429	1185	1220	1197	1630
1900000	1216	1445	1148	1201	1186	1355
1920000	1409	1412	1148	1248	1174	1321
1940000	1247	1414	1137	1314	1170	1373
1960000	1506	1208	1181	1207	1208	1387
1980000	1304	1439	1180	1217	1632	1378
2000000	1268	1219	1152	1266	1322	1297

Array Size- 3000000

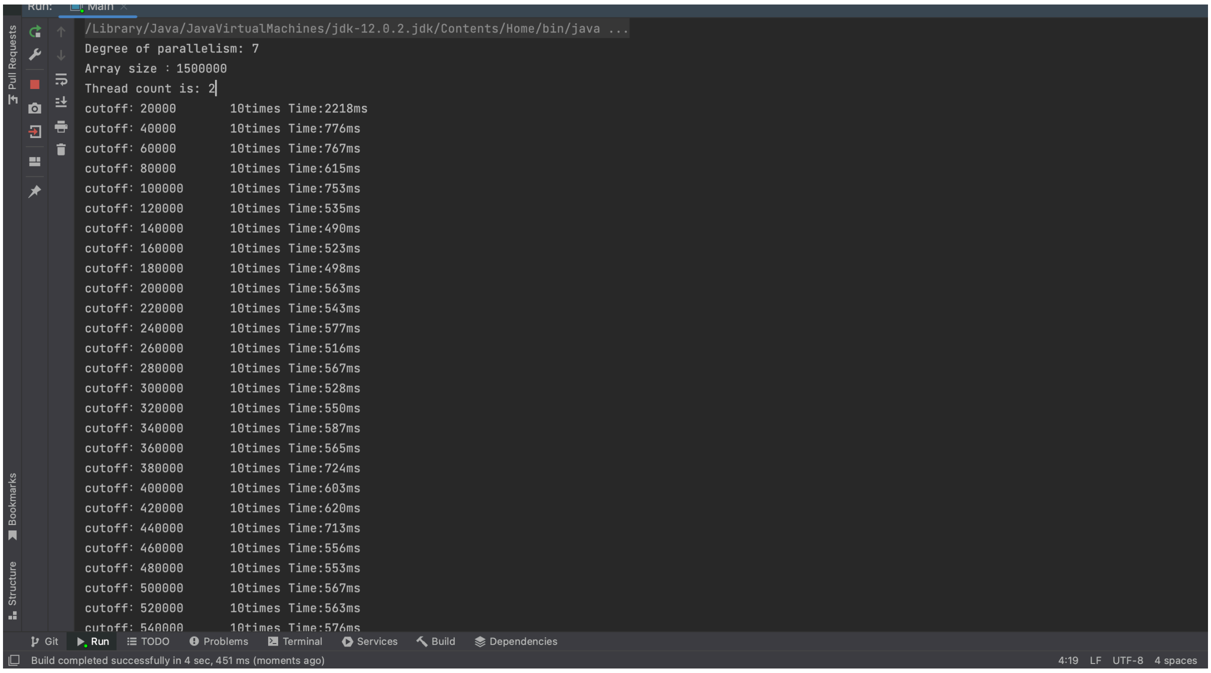


cutoff	T=2	T=4	T=8	T=16	T=32	T=64
20000	1702	1441	1705	1430	1676	1719
40000	1340	1390	1438	1319	1469	1397
60000	1475	1276	1169	1991	1246	1298
80000	1637	1186	1192	1282	1211	1369
100000	1332	1221	1454	1196	1176	1558
120000	1313	1259	1367	1285	1234	1660
140000	1167	1163	1286	1179	1228	1508
160000	1288	1134	1561	1138	1146	1503
180000	1232	1203	1461	1121	1260	1375
200000	1178	1330	1177	1165	1188	1265
220000	1214	1374	1404	1257	1271	1320
240000	1213	1921	1496	1302	1285	1337
260000	1347	1821	1562	1365	1335	1521
280000	1119	2157	1177	1155	1077	1163
300000	1266	1413	1358	1741	1161	1284
320000	1303	1802	1537	1261	1176	1483
340000	2016	1281	1762	1299	1272	1436
360000	1224	1170	1140	1157	1145	1144
380000	1369	1120	1403	1229	1212	1343

400000	1376	1219	1502	1269	1202	1334
420000	1444	1253	1293	1374	1254	1386
440000	1150	1116	1415	1133	1232	1538
460000	1206	1167	1124	1187	1122	1265
480000	1347	1418	1189	1164	1266	1155
500000	1265	1261	1233	1501	1182	1265
520000	1412	1487	1215	1336	1307	1260
540000	1126	1150	1254	1133	1258	1310
560000	1179	1331	1292	1141	1097	1277
580000	1319	1385	1125	1238	1248	1217
600000	1250	1267	1167	1149	1187	1257
620000	1297	1356	1225	1287	1291	1869
640000	1344	1208	1202	1258	1469	1305
660000	1276	1194	1233	1117	1098	1353
680000	1273	1302	1106	1187	1141	1152
700000	1171	1378	1219	1169	1264	1263
720000	1217	1223	1248	2052	1194	1274
740000	1326	1247	1142	1295	1324	1298
760000	1136	1129	1283	1153	1136	1447
780000	1166	1229	1137	1356	1148	1199
800000	1237	1527	1140	1388	1410	1179
820000	1412	1760	1203	1223	1222	1153
840000	1289	1246	1199	1207	1229	1222
860000	1250	1417	1161	1535	1230	1253
880000	1451	1727	1215	1346	1226	1240
900000	1470	1259	1243	1382	1302	1280
920000	1165	1148	1251	1163	1133	1400
940000	1159	1224	1282	1209	1145	1303
960000	1163	1202	1132	1210	1196	1174
980000	1175	1277	1135	1297	1254	1196
1000000	1151	1212	1163	1250	1214	1194
1020000	1325	1219	1186	1246	1235	1252
1040000	1273	1165	1280	1209	1203	1230
1060000	1263	1266	1218	1338	1322	1264
1080000	1113	1120	1249	1171	1132	1284
1100000	1206	1175	1262	1197	1130	1317
1120000	1425	1176	1125	1219	1161	1155
1140000	1204	1224	1146	1458	1166	1208
1160000	1231	1220	1171	1604	1211	1176
1180000	1273	1166	1207	1325	1190	1248
1200000	1190	1221	1179	1255	1189	1268
1220000	1273	1245	1219	1316	1253	1233

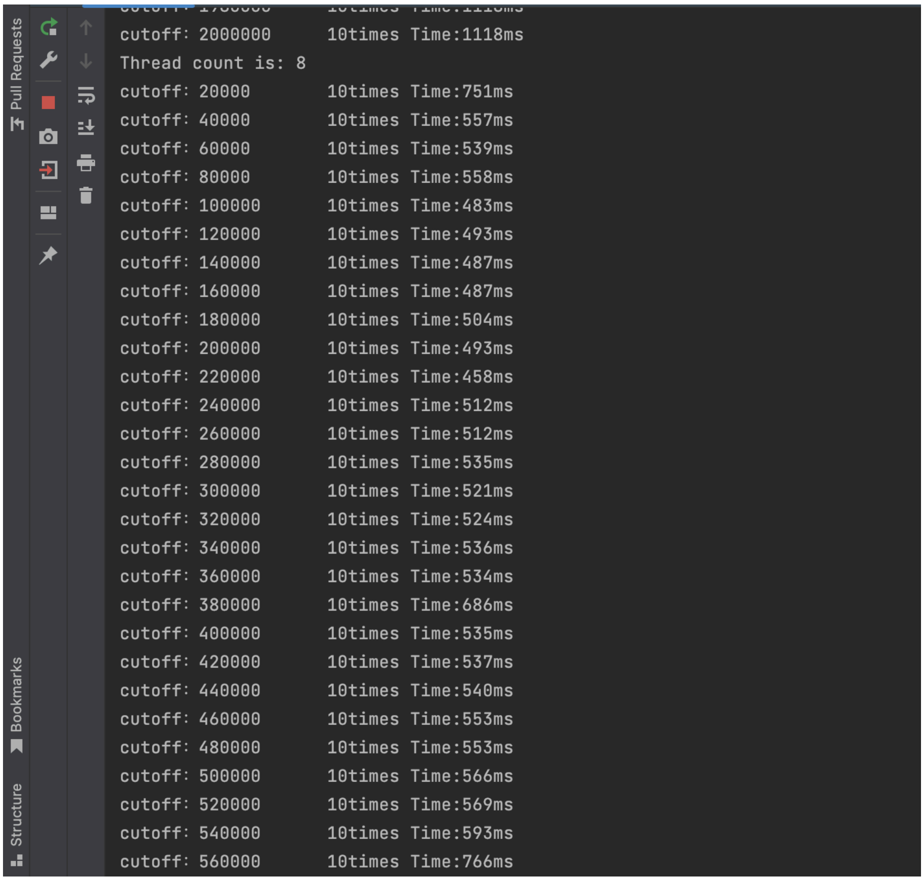
1240000	1191	1208	1672	1198	1264	1280
1260000	1263	1267	1373	1149	1121	1465
1280000	1186	1196	1173	1243	1152	1396
1300000	1284	1213	1138	1388	1182	1300
1320000	1274	1174	1171	1217	1275	1270
1340000	1278	1276	1242	1612	1177	1285
1360000	1345	1194	1539	1969	1213	1260
1380000	1366	1323	1278	1453	1170	1241
1400000	1182	1151	1297	1158	1367	1240
1420000	1185	1174	1398	1161	1124	1385
1440000	1173	1923	1151	1204	1154	1208
1460000	1288	1215	1129	1194	1152	1377
1480000	1449	1231	1196	1198	1195	1417
1500000	1284	1259	1220	1594	1140	1777
1520000	1550	1915	1479	1538	1510	1624
1540000	1551	1621	1476	1597	1482	1526
1560000	1419	1484	1540	1462	1432	1646
1580000	1524	1808	1777	1597	1443	1520
1600000	1411	1424	2002	1438	1433	1511
1620000	1406	1401	1635	1390	1520	1488
1640000	1440	1436	1529	1423	1535	1473
1660000	1400	1418	1491	1397	1401	1761
1680000	1801	1415	1694	1425	1405	1424
1700000	1728	1408	1446	1450	1397	1415
1720000	1521	1403	1457	1435	1480	1461
1740000	1711	1491	1438	1506	1385	1463
1760000	1708	1432	1420	1494	1389	2281
1780000	1564	1424	1455	1443	1439	1469
1800000	1544	1404	1654	1444	1452	1429
1820000	1480	1434	1518	1425	1596	1761
1840000	1477	1439	1493	1474	1741	1622
1860000	1456	1406	1477	1461	1703	1497
1880000	1495	1552	1523	1445	1738	1488
1900000	1451	1980	1500	1469	1480	1560
1920000	1641	1527	1523	1556	1479	1611
1940000	1422	1488	1737	1406	1574	1478
1960000	1408	1449	1573	1409	1432	1807
1980000	1418	1423	1671	1431	1494	1674
2000000	1446	1431	1642	1690	1841	1533

Unit Test Screenshots:



```
/Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
Array size : 1500000
Thread count is: 2
cutoff: 20000       10times Time:2218ms
cutoff: 40000       10times Time:776ms
cutoff: 60000       10times Time:767ms
cutoff: 80000       10times Time:615ms
cutoff: 100000      10times Time:753ms
cutoff: 120000      10times Time:535ms
cutoff: 140000      10times Time:490ms
cutoff: 160000      10times Time:523ms
cutoff: 180000      10times Time:498ms
cutoff: 200000      10times Time:563ms
cutoff: 220000      10times Time:543ms
cutoff: 240000      10times Time:577ms
cutoff: 260000      10times Time:516ms
cutoff: 280000      10times Time:567ms
cutoff: 300000      10times Time:528ms
cutoff: 320000      10times Time:550ms
cutoff: 340000      10times Time:587ms
cutoff: 360000      10times Time:565ms
cutoff: 380000      10times Time:724ms
cutoff: 400000      10times Time:603ms
cutoff: 420000      10times Time:620ms
cutoff: 440000      10times Time:713ms
cutoff: 460000      10times Time:556ms
cutoff: 480000      10times Time:553ms
cutoff: 500000      10times Time:567ms
cutoff: 520000      10times Time:563ms
cutoff: 540000      10times Time:576ms
```

Build completed successfully in 4 sec, 451 ms (moments ago)



```
cutoff: 2000000     10times Time:1118ms
Thread count is: 8
cutoff: 20000       10times Time:751ms
cutoff: 40000       10times Time:557ms
cutoff: 60000       10times Time:539ms
cutoff: 80000       10times Time:558ms
cutoff: 100000      10times Time:483ms
cutoff: 120000      10times Time:493ms
cutoff: 140000      10times Time:487ms
cutoff: 160000      10times Time:487ms
cutoff: 180000      10times Time:504ms
cutoff: 200000      10times Time:493ms
cutoff: 220000      10times Time:458ms
cutoff: 240000      10times Time:512ms
cutoff: 260000      10times Time:512ms
cutoff: 280000      10times Time:535ms
cutoff: 300000      10times Time:521ms
cutoff: 320000      10times Time:524ms
cutoff: 340000      10times Time:536ms
cutoff: 360000      10times Time:534ms
cutoff: 380000      10times Time:686ms
cutoff: 400000      10times Time:535ms
cutoff: 420000      10times Time:537ms
cutoff: 440000      10times Time:540ms
cutoff: 460000      10times Time:553ms
cutoff: 480000      10times Time:553ms
cutoff: 500000      10times Time:566ms
cutoff: 520000      10times Time:569ms
cutoff: 540000      10times Time:593ms
cutoff: 560000      10times Time:766ms
```