Program Structures and Algorithms

Spring 2023(SEC –01)

NAME: Anay Rajesh Pampatwar

NUID: 002766273

**Task:** To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. We will consider two different schemes for deciding whether to sort in parallel.

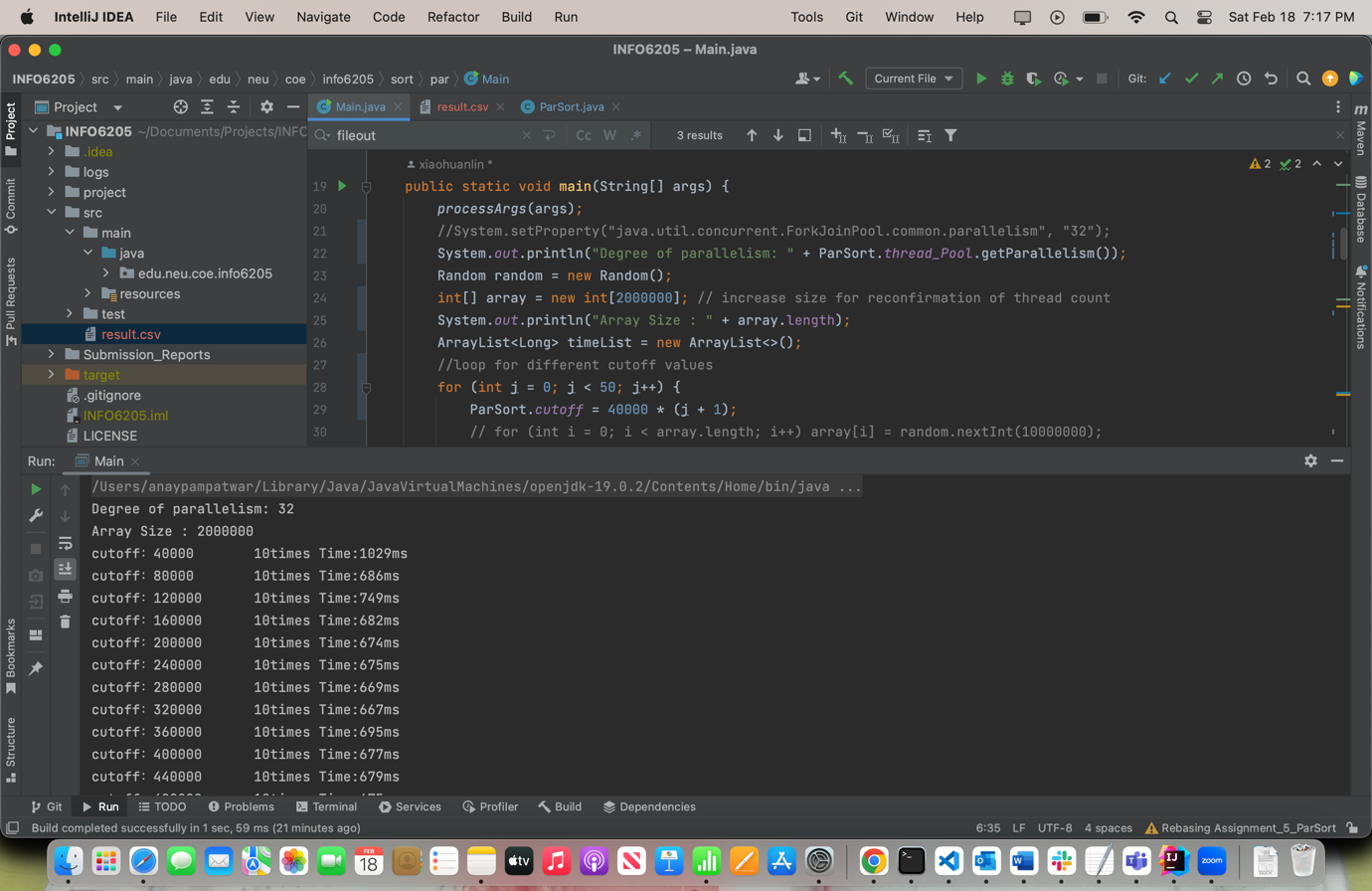
* A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
* Recursion 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).

An appropriate combination of these is to be found.

**Relationship Conclusion:** From the experiments that I carried out, a cutoff value of **100000** and thread count of **8** is ideal for the algorithm to work most effectively

**Evidence to support that conclusion:**

I am using an array of size 2000000 to sort. The cutoff values are incremented by 40000 from 40000 to 2000000. Graph is created for threads 2,4,8,16,32.



Graphical user interface, text

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Degree of parallelism: 2 | Degree of parallelism: 4 | Degree of parallelism: 8 | Degree of parallelism: 16 | Degree of parallelism: 32 |
| cutoff：40000 | 1119 | 1082 | 1092 | 1180 | 1029 |
| cutoff：80000 | 913 | 733 | 735 | 720 | 686 |
| cutoff：120000 | 919 | 744 | 716 | 690 | 749 |
| cutoff：160000 | 976 | 755 | 692 | 780 | 682 |
| cutoff：200000 | 952 | 768 | 674 | 807 | 674 |
| cutoff：240000 | 944 | 822 | 683 | 693 | 675 |
| cutoff：280000 | 1026 | 798 | 697 | 684 | 669 |
| cutoff：320000 | 1039 | 795 | 680 | 686 | 667 |
| cutoff：360000 | 1031 | 796 | 683 | 675 | 695 |
| cutoff：400000 | 1023 | 791 | 687 | 696 | 677 |
| cutoff：440000 | 1022 | 784 | 674 | 693 | 679 |
| cutoff：480000 | 1011 | 794 | 739 | 692 | 675 |
| cutoff：520000 | 1101 | 755 | 792 | 762 | 752 |
| cutoff：560000 | 1103 | 756 | 765 | 757 | 756 |
| cutoff：600000 | 1089 | 755 | 764 | 760 | 754 |
| cutoff：640000 | 1098 | 760 | 766 | 759 | 758 |
| cutoff：680000 | 1107 | 757 | 764 | 761 | 756 |
| cutoff：720000 | 1170 | 753 | 775 | 764 | 763 |
| cutoff：760000 | 1115 | 755 | 789 | 760 | 755 |
| cutoff：800000 | 1107 | 763 | 756 | 766 | 755 |
| cutoff：840000 | 1100 | 761 | 757 | 763 | 758 |
| cutoff：880000 | 1097 | 777 | 755 | 763 | 758 |
| cutoff：920000 | 1111 | 760 | 756 | 756 | 755 |
| cutoff：960000 | 1130 | 757 | 757 | 759 | 758 |
| cutoff：1000000 | 1103 | 763 | 761 | 760 | 752 |
| cutoff：1040000 | 1053 | 1049 | 1056 | 1055 | 1052 |
| cutoff：1080000 | 1057 | 1054 | 1058 | 1053 | 1054 |
| cutoff：1120000 | 1054 | 1045 | 1053 | 1062 | 1054 |
| cutoff：1160000 | 1050 | 1043 | 1059 | 1058 | 1061 |
| cutoff：1200000 | 1053 | 1040 | 1065 | 1057 | 1055 |
| cutoff：1240000 | 1065 | 1044 | 1055 | 1058 | 1052 |
| cutoff：1280000 | 1060 | 1051 | 1064 | 1057 | 1054 |
| cutoff：1320000 | 1072 | 1050 | 1056 | 1059 | 1053 |
| cutoff：1360000 | 1063 | 1059 | 1053 | 1059 | 1055 |
| cutoff：1400000 | 1057 | 1052 | 1056 | 1062 | 1056 |
| cutoff：1440000 | 1058 | 1054 | 1057 | 1057 | 1056 |
| cutoff：1480000 | 1055 | 1087 | 1057 | 1055 | 1057 |
| cutoff：1520000 | 1047 | 1109 | 1061 | 1065 | 1059 |
| cutoff：1560000 | 1058 | 1125 | 1056 | 1060 | 1058 |
| cutoff：1600000 | 1057 | 1050 | 1056 | 1062 | 1059 |
| cutoff：1640000 | 1058 | 1051 | 1055 | 1061 | 1053 |
| cutoff：1680000 | 1065 | 1055 | 1057 | 1055 | 1058 |
| cutoff：1720000 | 1064 | 1051 | 1054 | 1059 | 1057 |
| cutoff：1760000 | 1071 | 1047 | 1059 | 1059 | 1059 |
| cutoff：1800000 | 1056 | 1050 | 1058 | 1053 | 1091 |
| cutoff：1840000 | 1056 | 1047 | 1057 | 1059 | 1061 |
| cutoff：1880000 | 1054 | 1047 | 1072 | 1062 | 1062 |
| cutoff：1920000 | 1052 | 1044 | 1067 | 1058 | 1054 |
| cutoff：1960000 | 1049 | 1046 | 1059 | 1056 | 1102 |
| cutoff：2000000 | 1052 | 1048 | 1056 | 1056 | 1509 |

**Graphical Representation:**

Array Size : 2000000

cutoff values are incremented by 40000 from 40000 to 2000000

**Chart, line chart

Description automatically generated**

From above graph it is evident that ideal cutoff value is 100000.

Let us confirm this by using an array of length 400000. Below are the results:

Graphical user interface, text

Description automatically generated

Degree of parallelism: 8

Array Size : 4000000

cutoff：40000 10times Time:1939ms

cutoff：80000 10times Time:1322ms

cutoff：120000 10times Time:1341ms

cutoff：160000 10times Time:1294ms

cutoff：200000 10times Time:1393ms

cutoff：240000 10times Time:1322ms

cutoff：280000 10times Time:1312ms

cutoff：320000 10times Time:1325ms

cutoff：360000 10times Time:1407ms

cutoff：400000 10times Time:1320ms

cutoff：440000 10times Time:1301ms

cutoff：480000 10times Time:1277ms

cutoff：520000 10times Time:1278ms

cutoff：560000 10times Time:1278ms

cutoff：600000 10times Time:1269ms

cutoff：640000 10times Time:1249ms

cutoff：680000 10times Time:1267ms

cutoff：720000 10times Time:1249ms

cutoff：760000 10times Time:1266ms

cutoff：800000 10times Time:1250ms

cutoff：840000 10times Time:1259ms

cutoff：880000 10times Time:1232ms

cutoff：920000 10times Time:1322ms

cutoff：960000 10times Time:1237ms

cutoff：1000000 10times Time:1273ms

cutoff：1040000 10times Time:1459ms

cutoff：1080000 10times Time:1452ms

cutoff：1120000 10times Time:1453ms

cutoff：1160000 10times Time:1455ms

cutoff：1200000 10times Time:1445ms

cutoff：1240000 10times Time:1468ms

cutoff：1280000 10times Time:1451ms

cutoff：1320000 10times Time:1491ms

cutoff：1360000 10times Time:1566ms

cutoff：1400000 10times Time:1465ms

cutoff：1440000 10times Time:1460ms

cutoff：1480000 10times Time:1456ms

cutoff：1520000 10times Time:1456ms

cutoff：1560000 10times Time:1452ms

cutoff：1600000 10times Time:1451ms

cutoff：1640000 10times Time:1455ms

cutoff：1680000 10times Time:1471ms

cutoff：1720000 10times Time:1449ms

cutoff：1760000 10times Time:1456ms

cutoff：1800000 10times Time:1459ms

cutoff：1840000 10times Time:1483ms

cutoff：1880000 10times Time:1456ms

cutoff：1920000 10times Time:1483ms

cutoff：1960000 10times Time:1457ms

cutoff：2000000 10times Time:1452ms

Process finished with exit code 0

Graph:

For Degree of parallelism: 8

Array Size : 4000000

cutoff values are incremented by 40000 from 40000 to 2000000

Chart, line chart

Description automatically generated



The point marked in circle in above graph is for cutoff 100000.

So to conclude a cutoff value of 100000 and thread count of 8 is ideal for the algorithm to work most effectively