

Program Structures and Algorithms
Spring 2023(Section - 1)

Name: Mani Charan Reddy Loka

NUID: 002727403

Task:

- 1) A cutoff (defaults to, say, 1000) which you will update
- 2) Recursion depth or the number of available threads.

Relationship Conclusion:

By running the experiments for different combinations of cutoff values, threads, and array sizes, sufficient data is generated to conclude the optimal number of threads required for the algorithm. Upon analyzing the same, it can be seen that using 4 threads gives the best performance while increasing it any further does not have a significant improvement in the performance as the cost of thread creation and synchronization might offset the work done by parallelism.

On average, the lowest runtime is achieved when the cutoff value is 30% of the array size.

The relationship between thread count(t) and recursion depth d is $t=2^d$

Evidence to Conclusion:

Below are the runtimes in ms for different combinations of array size, threads, and cutoffs

ArraySize: 100000

Cutoff\No of Threads	2	4	8	16	32	64
10000	293ms	123ms	85ms	66ms	61ms	67ms
20000	163ms	73ms	62ms	60ms	62ms	59ms
30000	141ms	104ms	66ms	60ms	63ms	63ms
40000	112ms	65ms	65ms	65ms	62ms	64ms
50000	97ms	65ms	63ms	61ms	62ms	62ms
60000	86ms	84ms	79ms	84ms	80ms	80ms
70000	87ms	84ms	83ms	83ms	84ms	79ms
80000	123ms	84ms	81ms	80ms	83ms	82ms
90000	88ms	79ms	80ms	79ms	81ms	82ms
100000	88ms	89ms	82ms	78ms	84ms	76ms

Program Structures and Algorithms Spring 2023(Section - 1)

eclipse-workspace - INFO6205Assignment/src/main/java/edu/neu/coe/info6205/sort/par/Main.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- edu.neu.coe.info6205.graphs.generic_BFS_an
- edu.neu.coe.info6205.graphs.gis
- edu.neu.coe.info6205.graphs.tunnels
- edu.neu.coe.info6205.graphs.undirected
- edu.neu.coe.info6205.greedy
- edu.neu.coe.info6205.lab_1
- edu.neu.coe.info6205.life.base
- edu.neu.coe.info6205.life.library
- edu.neu.coe.info6205.pq
- edu.neu.coe.info6205.randomwalk
- edu.neu.coe.info6205.reduction
- edu.neu.coe.info6205.runLengthEncoding
- edu.neu.coe.info6205.sort
- edu.neu.coe.info6205.sort.classic
- edu.neu.coe.info6205.sort.counting
- edu.neu.coe.info6205.sort.elementary
- edu.neu.coe.info6205.sort.hashCode
- edu.neu.coe.info6205.sort.linearithmic
- edu.neu.coe.info6205.sort.par
 - Main.java
 - ParSort.java
- edu.neu.coe.info6205.symbolTable
- edu.neu.coe.info6205.symbolTable.hashtable
- edu.neu.coe.info6205.symbolTable.tree
- edu.neu.coe.info6205.threesum
- edu.neu.coe.info6205.union_find
- edu.neu.coe.info6205.util
- src/main/resources
- src/test/java
- src/test/resources

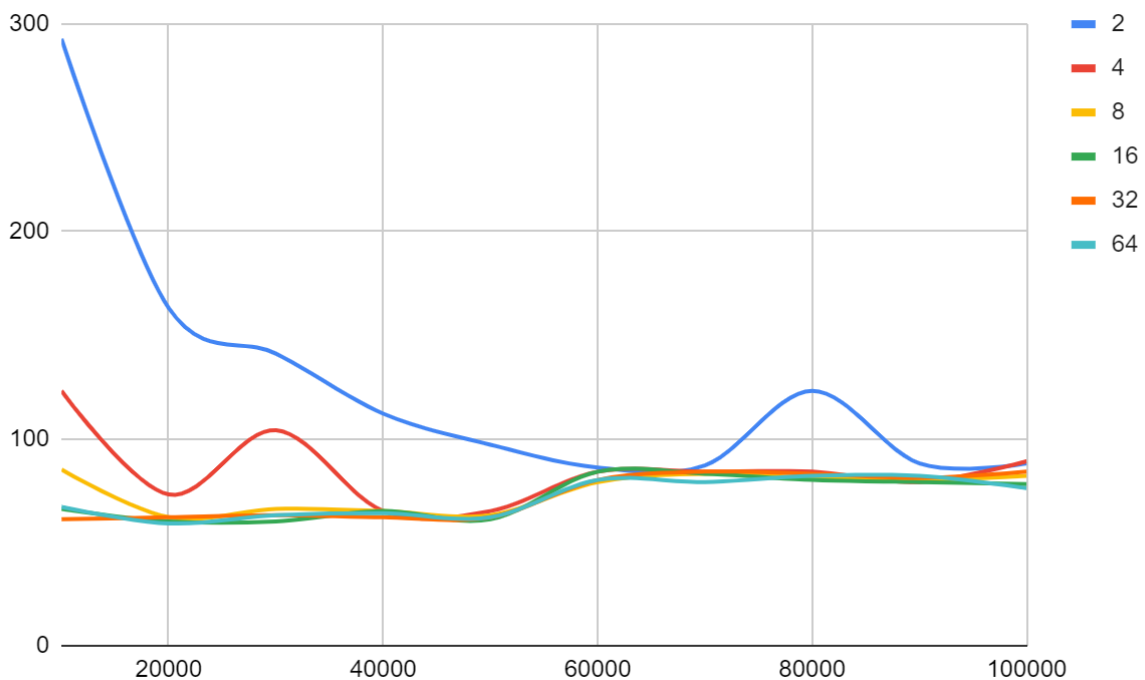
ParSort.java

```
26 int[] array = new int[100000];
27 ArrayList<Long> timeList = new ArrayList<>();
28 for (int j = 0; j < 10; j++) {
29     ParSort.cutoff = 10000 * (j + 1);
```

Console

```
<terminated> Main [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe (18-Feb-2023, 9:02:35 pm)
cutoff:60000 10times Time:84ms
cutoff:70000 10times Time:83ms
cutoff:80000 10times Time:80ms
cutoff:90000 10times Time:79ms
cutoff:100000 10times Time:78ms
Degree of parallelism: 32
cutoff:10000 10times Time:61ms
cutoff:20000 10times Time:62ms
cutoff:30000 10times Time:63ms
cutoff:40000 10times Time:62ms
cutoff:50000 10times Time:62ms
cutoff:60000 10times Time:80ms
cutoff:70000 10times Time:84ms
cutoff:80000 10times Time:83ms
cutoff:90000 10times Time:81ms
cutoff:100000 10times Time:84ms
Degree of parallelism: 64
cutoff:10000 10times Time:67ms
cutoff:20000 10times Time:59ms
cutoff:30000 10times Time:63ms
cutoff:40000 10times Time:64ms
cutoff:50000 10times Time:62ms
cutoff:60000 10times Time:80ms
cutoff:70000 10times Time:79ms
cutoff:80000 10times Time:82ms
cutoff:90000 10times Time:82ms
cutoff:100000 10times Time:76ms
```

Time in ms vs cut-off graph for array size of 100000.



Program Structures and Algorithms Spring 2023(Section - 1)

Below are the values for array size 250000

Cutoff\No of Threads	2	4	8	16	32	64
10000	489.00	263.00	178.00	181.00	131.00	171.00
20000	389.00	158.00	137.00	135.00	126.00	144.00
30000	241.00	158.00	134.00	129.00	151.00	129.00
40000	199.00	159.00	143.00	131.00	127.00	140.00
50000	231.00	151.00	139.00	147.00	130.00	147.00
60000	198.00	154.00	134.00	131.00	123.00	135.00
70000	232.00	153.00	143.00	145.00	165.00	147.00
80000	332.00	158.00	145.00	148.00	168.00	144.00
90000	272.00	147.00	150.00	140.00	157.00	140.00
100000	243.00	147.00	149.00	138.00	190.00	143.00

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays a project structure for 'INFO6205Assignment [INFO6205Assignment Spring2023]'. The main editor shows a Java file 'Main.java' with the following code:

```

24 System.out.println("Degree of parallelism: " + pool.getParallelism());
25 Random random = new Random();
26 int[] array = new int[250000];

```

The Console window at the bottom shows the output of the program, which includes timing results for various cutoff values and the degree of parallelism for different thread counts:

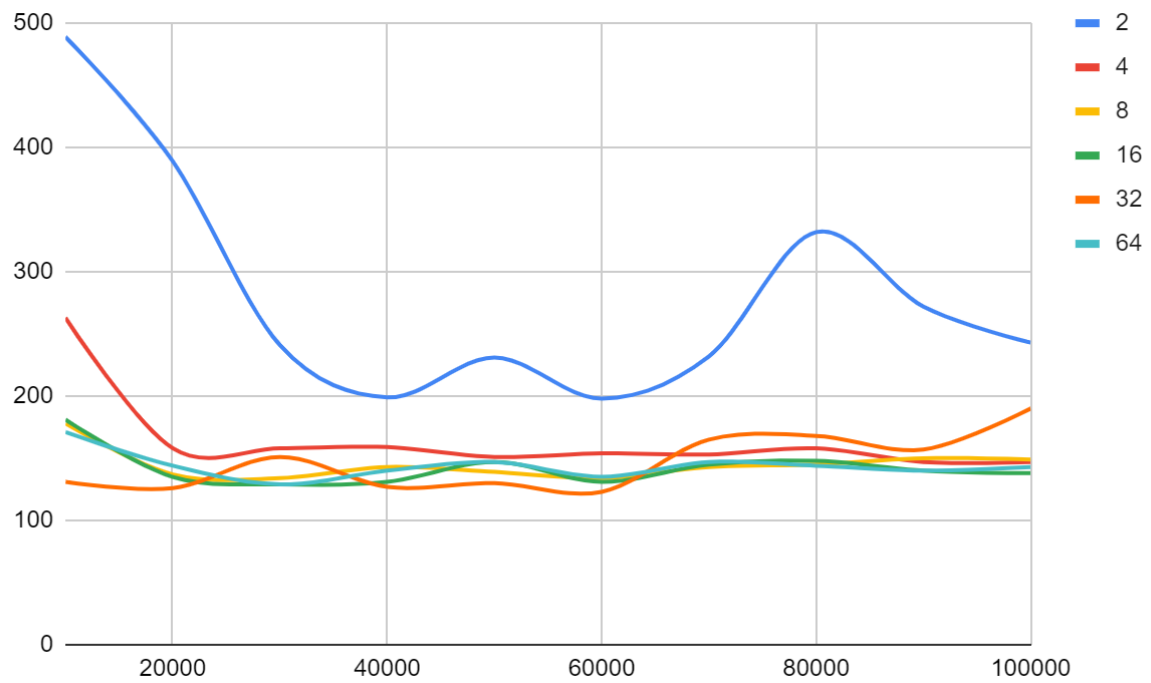
```

<terminated> Main [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220903-1038\jre\bin\javaw.exe (18-Feb-2023, 9:40:46 pm)
cutoff:50000 10times Time:147ms
cutoff:60000 10times Time:131ms
cutoff:70000 10times Time:145ms
cutoff:80000 10times Time:148ms
cutoff:90000 10times Time:140ms
cutoff:100000 10times Time:138ms
Degree of parallelism: 32
cutoff:10000 10times Time:131ms
cutoff:20000 10times Time:126ms
cutoff:30000 10times Time:151ms
cutoff:40000 10times Time:127ms
cutoff:50000 10times Time:130ms
cutoff:60000 10times Time:123ms
cutoff:70000 10times Time:165ms
cutoff:80000 10times Time:168ms
cutoff:90000 10times Time:157ms
cutoff:100000 10times Time:190ms
Degree of parallelism: 64
cutoff:10000 10times Time:171ms
cutoff:20000 10times Time:144ms
cutoff:30000 10times Time:129ms
cutoff:40000 10times Time:140ms
cutoff:50000 10times Time:147ms
cutoff:60000 10times Time:135ms
cutoff:70000 10times Time:147ms
cutoff:80000 10times Time:144ms
cutoff:90000 10times Time:140ms
cutoff:100000 10times Time:143ms

```

Program Structures and Algorithms
Spring 2023(Section - 1)

Time in ms vs cutoff graph for array size = 250000



Below are the values for array size = 500000

Cutoff/No of Threads	2	4	8	16	32	64
10000	961.00	396.00	306.00	315.00	361.00	295.00
20000	346.00	322.00	247.00	289.00	281.00	286.00
30000	368.00	280.00	253.00	289.00	285.00	277.00
40000	394.00	290.00	260.00	293.00	286.00	290.00
50000	353.00	291.00	282.00	304.00	290.00	298.00
60000	360.00	299.00	281.00	316.00	311.00	315.00
70000	417.00	305.00	275.00	311.00	314.00	311.00
80000	411.00	297.00	257.00	309.00	298.00	291.00
90000	410.00	296.00	259.00	301.00	299.00	289.00
100000	406.00	299.00	255.00	301.00	293.00	280.00

Program Structures and Algorithms

Spring 2023(Section - 1)

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays a project structure for 'INFO6205 Spring2023' with various sub-packages like 'edu.neu.coe.info6205.balsearchtree', 'edu.neu.coe.info6205.bqs', etc. The main editor shows a Java file 'Main.java' with the following code:

```
24 System.out.println("Degree of parallelism: " + pool.getParallelism());
25 Random random = new Random();
26 int[] array = new int[500000];
27 Arrays.sort(array, random);
```

The Console on the right shows the output of the program, which includes a table of execution times for different cutoff values and degrees of parallelism:

Cutoff	10times Time
50000	304ms
60000	316ms
70000	311ms
80000	309ms
90000	301ms
100000	302ms

Below this, the output shows the 'Degree of parallelism' and the 'cutoff' value for each degree of parallelism (32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, 4194304, 8388608, 16777216, 33554432, 67108864, 134217728, 268435456, 536870912, 1073741824, 2147483648, 4294967296, 8589934592, 17179869184, 34359738368, 68719476736, 137438953472, 274877906944, 549755813888, 1099511627776, 2199023255552, 4398046511104, 8796093022208, 17592186044416, 35184372088832, 70368744177664, 140737488355328, 281474976710656, 562949953421312, 1125899906842624, 2251799813685248, 4503599627370496, 9007199254740992, 18014398509481984, 36028797018963968, 72057594037927936, 144115188075855872, 288230376151711744, 576460752303423488, 1152921504606846976, 2305843009213693952, 4611686018427387904, 9223372036854775808, 18446744073709551616, 36893488147419103232, 73786976294838206464, 147573952589676412928, 295147905179352825856, 590295810358705651712, 1180591620717411303424, 2361183241434822606848, 4722366482869645213696, 9444732965739290427392, 18889465931478580854784, 37778931862957161709568, 75557863725914323419136, 151115727451828646838272, 302231454903657293676544, 604462909807314587353088, 1208925819614629174706176, 2417851639229258349412352, 4835703278458516698824704, 9671406556917033397649408, 19342813113834066795298816, 38685626227668133590597632, 77371252455336267181195264, 154742504910672534362390528, 309485009821345068724781056, 618970019642690137449562112, 1237940039285380274899124224, 2475880078570760549798248448, 4951760157141521099596496896, 9903520314283042199192993792, 19807040628566084398385987584, 39614081257132168796771975168, 79228162514264337593543950336, 158456325028528675187087900672, 316912650057057350374175801344, 633825300114114700748351602688, 1267650600228229401496703205376, 2535301200456458802993406410752, 5070602400912917605986812821504, 10141204801825835211973625643008, 20282409603651670423947251286016, 40564819207303340847894502572032, 81129638414606681695789005144064, 162259276829213363391578010288128, 324518553658426726783156020576256, 649037107316853453566312041152512, 1298074214633706907132624082305024, 2596148429267413814265248164610048, 5192296858534827628530496329220096, 10384593717069655257060992658440192, 20769187434139310514121985316880384, 41538374868278621028243970633760768, 83076749736557242056487941267521536, 166153499473114484112975882535043072, 332306998946228968225951765070086144, 664613997892457936451903530140172288, 1329227995784915872903807060280344576, 2658455991569831745807614120560689152, 5316911983139663491615228241121378304, 10633823966279326983230456482242756608, 21267647932558653966460912964485513216, 42535295865117307932921825928971026432, 85070591730234615865843651857942052864, 170141183460469231731687303715884105728, 340282366920938463463374607431768211456, 680564733841876926926749214863536422912, 1361129467683753853853498429727072845824, 2722258935367507707706996859454145691648, 5444517870735015415413993718908291383296, 10889035741470030830827987437816582766592, 21778071482940061661655974875633165533184, 43556142965880123323311949751266331066368, 87112285931760246646623899502532662132736, 174224571863520493293247799005065324265472, 348449143727040986586495598010130648530944, 696898287454081973172991196020261297061888, 1393796574908163946345982392040522594123776, 2787593149816327892691964784081045188247552, 5575186299632655785383929568162090376495104, 11150372599265311570767859136324180752990208, 22300745198530623141535718272648361505980416, 44601490397061246283071436545296723011960832, 89202980794122492566142873090593446023921664, 178405961588244985132285746181186892047843328, 356811923176489970264571492362373784095686656, 713623846352979940529142984724747568191373312, 1427247692705959881058285969449495136382746624, 2854495385411919762116571938898990272765493248, 5708990770823839524233143877797980545530986496, 11417981541647679048466287755595961091061972992, 22835963083295358096932575511191922182123945984, 45671926166590716193865151022383844364247891968, 91343852333181432387730302044767688728495783936, 182687704666362864775460604089535377456991567872, 365375409332725729550921208179070754913983135744, 730750818665451459101842416358141509827966271488, 1461501637330902918203684832716283019655932542976, 2923003274661805836407369665432566039311865085952, 5846006549323611672814739330865132078623730171904, 11692013098647223345629478661730264157247460343808, 23384026197294446691258957323460528314494920687616, 46768052394588893382517914646921056628989841375232, 93536104789177786765035829293842113257979682750464, 187072209578355573530071658587684226515959365500928, 374144419156711147060143317175368453031918731001856, 748288838313422294120286634350736906063837462003712, 1496577676626844588240573268701473812127674924007424, 2993155353253689176481146537402947624255349848014848, 5986310706507378352962293074805895248510699696029696, 11972621413014756705924586149611790497021399392059392, 23945242826029513411849172299223580994042798784118784, 47890485652059026823698344598447161988085597568237568, 95780971304118053647396689196894323976171195136475136, 191561942608236107294793378393788647952342390272950272, 383123885216472214589586756787577295904684780545900544, 766247770432944429179173513575154591809369561091801088, 1532495540865888858358347027150309183618739122183602176, 3064991081731777716716694054300618367237478244367204352, 6129982163463555433433388108601236734474956488734408704, 12259964326927110866866776217202473468949912977468817408, 24519928653854221733733552434404946937899825954937634816, 49039857307708443467467104868809893875799651909875269632, 98079714615416886934934209737619787751599303819750539264, 196159429230833773869868419475239575503198607639501078528, 392318858461667547739736838950479151006397215279002157056, 784637716923335095479473677900958302012794430558004314112, 1569275433846670190958947355801916604025588861116008628224, 3138550867693340381917894711603833208051177722232017256448, 6277101735386680763835789423207666416102355444464034512896, 12554203470773361527671578846415332832204710888928069025792, 25108406941546723055343157692830665664409421777856138051584, 50216813883093446110686315385661331328818843555712276103168, 100433627766186892221372630771322662657637687111424552206336, 200867255532373784442745261542645325315275374222849104412672, 401734511064747568885490523085290650630550748445698208825344, 803469022129495137770981046170581301261101496891396417650688, 1606938044258990275541962092341162602522202993782792835301376, 3213876088517980551083924184682325205044405987565585670602752, 6427752177035961102167848369364650410088811975131171341205504, 12855504354071922204335696738729300820177623950262342682411008, 25711008708143844408671393477458601640355247900524685364822016, 51422017416287688817342786954917203280710495801049370729644032, 102844034832575377634685573909834406561420991602098741459288064, 205688069665150755269371147819668813122841983204197482918576128, 411376139330301510538742295639337626245683966408394965837152256, 822752278660603021077484591278675252491367932816789931674304512, 1645504557321206042154969182557350504982735865633579863348609024, 3291009114642412084309938365114701009965471731267159726697218048, 6582018229284824168619876730229402019930943462534319453394436096, 13164036458569648337239753460458804039861886925068638906788872192, 26328072917139296674479506920917608079723773850137277813577744384, 52656145834278593348959013841835216159447547700274555627155488768, 105312291668557186697918027683670432318895095400549111254310977536, 210624583337114373395836055367340864637790190801098222508621955072, 421249166674228746791672110734681729275580381602196445017243910144, 842498333348457493583344221469363458551160763204392890034487820288, 1684996666696914987166688442938726917102321526408785780068975640576, 3369993333393829974333376885877453834204643052817571560137951281152, 6739986666787659948666753771754907668409286105635143120275902562304, 13479973333575319897333507543509815336818572211270286240551805124608, 26959946667150639794667015087019630673637144422540572481103610249216, 53919893334301279589334030174039261347274288845081144962207220498432, 107839786668602559178668060348078522694548577690162289924414440996864, 215679573337205118357336120696157045389097155380324579848828881993728, 431359146674410236714672241392314090778194310760649159697657763987456, 862718293348820473429344482784628181556388621521298319395315527974912, 1725436586697640946858688965569256363112777243042596638790631055949824, 3450873173395281893717377931138512726225554486085193277581262111899648, 6901746346790563787434755862277025452451108972170386555162524223799296, 13803492693581127574869511724554050904902217944340773110325048447598592, 27606985387162255149739023449108101809804435888681546220650096895197184, 55213970774324510299478046898216203619608871777363092441300193790394368, 110427941548649020598956093796432407239217743554726184882600387580788736, 220855883097298041197912187592864814478435487109452369765200775161577472, 441711766194596082395824375185729628956870974218904739530401550323154944, 883423532389192164791648750371459257913741948437809479060803100646309888, 1766847064778384329583297500742918515827483896875618958121606201292619776, 3533694129556768659166595001485837031654967793751237916243212402585239552, 7067388259113537318333190002971674063309935587502475832486424805170479104, 14134776518227074636666380005943348126619871175004951664972849610340958208, 28269553036454149273332760011886696253239742350009903329945699220681916416, 56539106072908298546665520023773392506479484700019806659891398441363832832, 113078212145816597093331040047546785012958969400039613319782796882727665664, 226156424291633194186662080095093570025917938800079226639565593765455331328, 452312848583266388373324160190187140051835877600158453279131187530910662656, 904625697166532776746648320380374280103671755200316906558262375061821325312, 1809251394333065553493296640760748560207343510400633813116524750123642650624, 3618502788666131106986593281521497120414687020801267626233049500247285301248, 7237005577332262213973186563042994240829374041602535252466099000494570602496, 14474011154664524427946373126085988481658748083205070504932198000989141204992, 28948022309329048855892746252171976963317496166410141009864396001978282409984, 57896044618658097711785492504343953926634992332820282019728792003956564819968, 115792089237316195423570985008687907853269984665640564039457584007913129639936, 231584178474632390847141970017375815706539969331281128078915168015826259279872, 463168356949264781694283940034751631413079938662562256157830336031652518559744, 926336713898529563388567880069503262826159877325124512315660672063305037119488, 1852673427797059126777135760139006525652319754650249024631321344126610074238976, 3705346855594118253554271520278013051304639509300498049262642688253220148477952, 7410693711188236507108543040556026102609279018600996098525285376506440296955904, 14821387422376473014217086081112052205218558037201992197050570753012880593911808, 29642774844752946028434172162224104410437116074403984394101141506025761187823616, 59285549689505892056868344324448208820874232148807968788202283012051522375647232, 118571099379011784113736688648896417641748464297615937576404566024103044751294464, 237142198758023568227473377297792835283496928595231875152809132048206089502588928, 474284397516047136454946754595585670566993857190463750305618264096412179005177856, 948568795032094272909893509191171341133987714380927500611236528192824358010355712, 1897137590064188545819787018382342682267975428761855001222473056385648716020711424, 3794275180128377091639574036764685364535950857523710002444946112771297432041422848, 7588550360256754183279148073529370729071901715047420004889892225542594864082845696, 15177100720513508366558296147058741458143803430094840009779

Program Structures and Algorithms Spring 2023(Section - 1)

Below are the values for array size = 1000000

Cutoff\No of Threads	2	4	8	16	32	64
10000	1,306.00	773.00	711.00	605.00	611.00	699.00
20000	1,095.00	612.00	577.00	572.00	520.00	619.00
30000	627.00	654.00	656.00	663.00	537.00	628.00
40000	678.00	613.00	578.00	607.00	521.00	760.00
50000	647.00	613.00	591.00	582.00	501.00	737.00
60000	649.00	619.00	549.00	703.00	507.00	712.00
70000	707.00	650.00	508.00	746.00	537.00	687.00
80000	718.00	643.00	536.00	546.00	529.00	636.00
90000	737.00	650.00	515.00	502.00	506.00	728.00
100000	731.00	628.00	515.00	502.00	510.00	689.00

The screenshot shows the Eclipse IDE with a Java project named 'INFO6205Assignment'. The main class is 'Main.java', which contains the following code:

```

25 Random random = new Random();
26 int[] array = new int[1000000];
27 ArrayList<Long> timeList = new ArrayList<>();
28 for (int j = 0; j < 10; j++) {

```

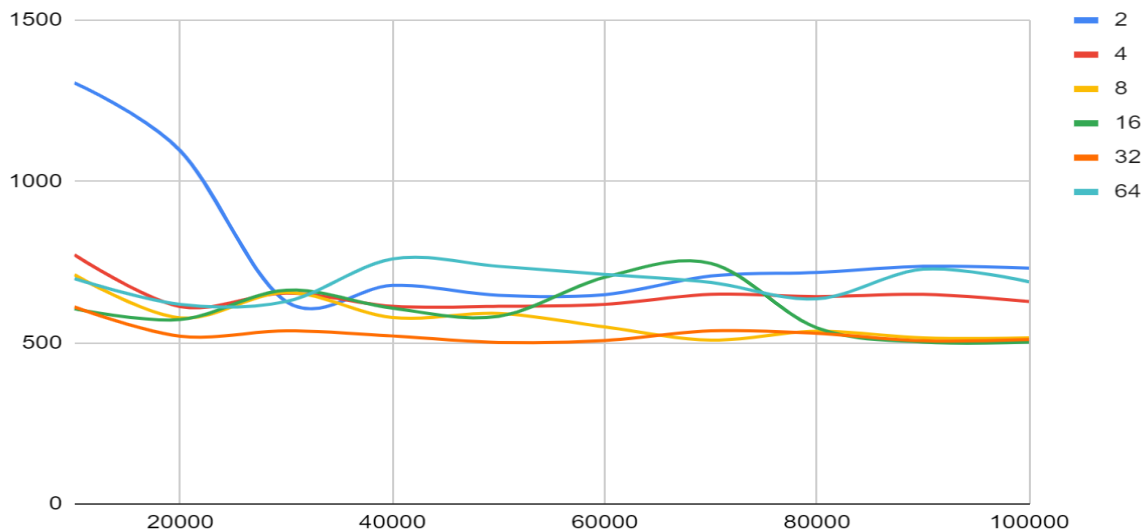
The console output shows the execution results for different cutoff values and degrees of parallelism:

```

<terminated> Main [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220903-1038\jre\bin\javaw.exe (18-Feb-2023, 9:58:56 pm --
cutoff:10000 10times Time:592ms
cutoff:60000 10times Time:703ms
cutoff:70000 10times Time:746ms
cutoff:80000 10times Time:546ms
cutoff:90000 10times Time:502ms
cutoff:100000 10times Time:502ms
Degree of parallelism: 32
cutoff:10000 10times Time:611ms
cutoff:20000 10times Time:520ms
cutoff:30000 10times Time:537ms
cutoff:40000 10times Time:521ms
cutoff:50000 10times Time:501ms
cutoff:60000 10times Time:507ms
cutoff:70000 10times Time:537ms
cutoff:80000 10times Time:529ms
cutoff:90000 10times Time:506ms
cutoff:100000 10times Time:510ms
Degree of parallelism: 64
cutoff:10000 10times Time:699ms
cutoff:20000 10times Time:619ms
cutoff:30000 10times Time:628ms
cutoff:40000 10times Time:760ms
cutoff:50000 10times Time:737ms
cutoff:60000 10times Time:712ms
cutoff:70000 10times Time:687ms
cutoff:80000 10times Time:636ms
cutoff:90000 10times Time:728ms
cutoff:100000 10times Time:689ms

```

Time in ms vs cutoff graph for array size = 1000000























Program Structures and Algorithms
Spring 2023(Section - 1)

The graphs show that increasing the thread count doesn't significantly improve the sort time of the parallel sort. The sort time is optimal when the value of thread is 4. Hence, using 4 threads is the optimal way of using parallel sorting.

Also, upon comparing the cutoff at which the which the 4 thread time is optimal we can find out that it is always optimal when the cutoff length is about 20~40% of the input length. Hence, we can average the value to around 30%.

Below is the screenshot of csv files generated for every different combination of array size and thread count

 ArraySize100000Threads32result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize100000Threads64result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads2result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads4result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads8result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads16result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads32result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize250000Threads64result.csv	18-02-2023 21:40	XLS Worksheet
 ArraySize500000Threads2result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize500000Threads4result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize500000Threads8result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize500000Threads16result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize500000Threads32result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize500000Threads64result.csv	18-02-2023 21:50	XLS Worksheet
 ArraySize1000000Threads2result.csv	18-02-2023 21:59	XLS Worksheet
 ArraySize1000000Threads4result.csv	18-02-2023 21:59	XLS Worksheet
 ArraySize1000000Threads8result.csv	18-02-2023 21:59	XLS Worksheet
 ArraySize1000000Threads16result.csv	18-02-2023 21:59	XLS Worksheet
 ArraySize1000000Threads32result.csv	18-02-2023 21:59	XLS Worksheet
 ArraySize1000000Threads64result.csv	18-02-2023 21:59	XLS Worksheet