Program Structures and Algorithms

Spring 2024

NAME: Kartikey Vijayakumar Hebbar

NUID: 002276938

GITHUB LINK:

**Task: Assignment 5 (Parallel Sorting)**

**Relationship Conclusion:**

Conclusion 1: As per the plotted data points in the comparison between cutoff ratio (cutoff size/size of the array) and time taken for the parallel sort to sort the array, the best ratio where the algorithm performed the best is at 0.75. Moreover, the algorithm performance was the best between the range 0.75 to 0.89 across all array sizes combined.

Conclusion 2: After performing tests on various thread sizes (2 to 16), it is found that for an array of size 1000000, the algorithm performs best on 2 threads. Whereas, in case of the array size increased to 16000000, the algorithm seems to perform better on 4 thread size.

**Evidence to support that conclusion:**

1. To find the optimal cutoff range considering different array sizes while keeping the number of threads constant.

* Threads: 2

Array sizes: Increased from 1 million to 16 million (using the doubling approach)

Cutoff Ratio: Cutoff value/Size of array – 0.5 to 1 (incrementing 0.1 each time)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cutoff/Array size** | **1 Million** | **2 Million** | **4 Million** | **8 Million** | **16 Million** |
| 0.5 | 778 | 998 | 2128 | 3580 | 7022 |
| 0.51 | 456 | 906 | 2012 | 4101 | 8721 |
| 0.52 | 434 | 896 | 2056 | 4204 | 8592 |
| 0.53 | 438 | 900 | 2006 | 4115 | 8478 |
| 0.54 | 435 | 904 | 1954 | 4213 | 8578 |
| 0.55 | 445 | 895 | 1946 | 4094 | 8489 |
| 0.56 | 444 | 905 | 2001 | 4097 | 8425 |
| 0.57 | 454 | 921 | 1957 | 4174 | 8535 |
| 0.58 | 454 | 950 | 1936 | 4218 | 8401 |
| 0.59 | 447 | 915 | 1942 | 4085 | 8458 |
| 0.6 | 496 | 934 | 2062 | 4102 | 8473 |
| 0.61 | 470 | 902 | 1982 | 4178 | 8575 |
| 0.62 | 441 | 919 | 1957 | 4123 | 8438 |
| 0.63 | 434 | 892 | 1963 | 4133 | 8424 |
| 0.64 | 440 | 894 | 1958 | 4073 | 8341 |
| 0.65 | 435 | 898 | 1978 | 4167 | 8507 |
| 0.66 | 435 | 891 | 1950 | 4082 | 8365 |
| 0.67 | 437 | 933 | 1947 | 4031 | 8436 |
| 0.68 | 434 | 929 | 2008 | 4100 | 8363 |
| 0.69 | 440 | 900 | 2040 | 4233 | 8489 |
| 0.7 | 435 | 912 | 1956 | 4103 | 8405 |
| 0.71 | 433 | 906 | 1952 | 4087 | 8418 |
| 0.72 | 435 | 966 | 1974 | 4084 | 8357 |
| 0.73 | 434 | 917 | 1958 | 4217 | 8412 |
| 0.74 | 437 | 909 | 1952 | 4070 | 8488 |
| 0.75 | 429 | 939 | 1950 | 4082 | 8860 |
| 0.76 | 438 | 927 | 2042 | 4082 | 8506 |
| 0.77 | 432 | 930 | 2023 | 4267 | 8361 |
| 0.78 | 431 | 926 | 1955 | 4118 | 8553 |
| 0.79 | 435 | 907 | 1951 | 4073 | 8298 |
| 0.8 | 430 | 893 | 1964 | 4070 | 8484 |
| 0.81 | 441 | 892 | 1935 | 4181 | 8303 |
| 0.82 | 432 | 901 | 1944 | 4101 | 8544 |
| 0.83 | 436 | 893 | 1936 | 4259 | 8283 |
| 0.84 | 440 | 898 | 1980 | 4156 | 8430 |
| 0.85 | 430 | 896 | 2013 | 4143 | 8271 |
| 0.86 | 432 | 914 | 1957 | 4075 | 8509 |
| 0.87 | 431 | 914 | 1955 | 4089 | 8369 |
| 0.88 | 438 | 888 | 1962 | 4172 | 8421 |
| 0.89 | 431 | 893 | 1948 | 4147 | 8689 |
| 0.9 | 433 | 902 | 1956 | 4067 | 8379 |
| 0.91 | 448 | 894 | 1935 | 4075 | 8497 |
| 0.92 | 457 | 923 | 1973 | 4172 | 8432 |
| 0.93 | 455 | 936 | 1994 | 4119 | 8470 |
| 0.94 | 462 | 921 | 1976 | 4180 | 8351 |
| 0.95 | 445 | 927 | 1948 | 4323 | 8425 |
| 0.96 | 443 | 892 | 1943 | 4377 | 8527 |
| 0.97 | 447 | 895 | 1947 | 4073 | 8495 |
| 0.98 | 449 | 905 | 1949 | 4107 | 8321 |
| 0.99 | 433 | 899 | 1926 | 4106 | 8528 |
| 1 | 432 | 906 | 1959 | 4326 | 8382 |

A graph of different colored lines

Description automatically generated

* Threads: 4

Array sizes: Increased from 1 million to 16 million (using the doubling approach)

Cutoff Ratio: Cutoff value/Size of array – 0.5 to 1 (incrementing 0.1 each time)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cutoff/Array size** | **1 Million** | **2 Million** | **4 Million** | **8 Million** | **16 Million** |
| 0.5 | 771 | 1087 | 1834 | 3508 | 6570 |
| 0.51 | 517 | 937 | 2030 | 4167 | 8655 |
| 0.52 | 463 | 932 | 1980 | 4189 | 8139 |
| 0.53 | 458 | 927 | 1987 | 4123 | 8361 |
| 0.54 | 478 | 929 | 1971 | 4077 | 8097 |
| 0.55 | 483 | 946 | 1989 | 4085 | 8238 |
| 0.56 | 453 | 969 | 2064 | 4177 | 8090 |
| 0.57 | 452 | 966 | 2025 | 4226 | 8164 |
| 0.58 | 450 | 978 | 1987 | 4084 | 8083 |
| 0.59 | 455 | 931 | 1998 | 4122 | 8570 |
| 0.6 | 458 | 925 | 1989 | 4296 | 8985 |
| 0.61 | 452 | 944 | 2013 | 4127 | 9970 |
| 0.62 | 456 | 931 | 2034 | 4079 | 9250 |
| 0.63 | 472 | 936 | 2026 | 4104 | 8409 |
| 0.64 | 483 | 1064 | 2118 | 4263 | 8990 |
| 0.65 | 479 | 1011 | 2332 | 4067 | 8729 |
| 0.66 | 474 | 978 | 2060 | 4098 | 8739 |
| 0.67 | 459 | 931 | 2016 | 4030 | 8530 |
| 0.68 | 474 | 935 | 2011 | 4253 | 8622 |
| 0.69 | 459 | 931 | 2038 | 4101 | 8499 |
| 0.7 | 450 | 925 | 1987 | 4078 | 8586 |
| 0.71 | 449 | 941 | 2010 | 4127 | 8692 |
| 0.72 | 453 | 950 | 2070 | 4193 | 8519 |
| 0.73 | 449 | 977 | 1987 | 4072 | 8538 |
| 0.74 | 447 | 943 | 1996 | 4077 | 8552 |
| 0.75 | 451 | 959 | 2005 | 4142 | 8638 |
| 0.76 | 452 | 933 | 1986 | 4168 | 8478 |
| 0.77 | 451 | 933 | 2002 | 4044 | 8718 |
| 0.78 | 447 | 927 | 1986 | 4049 | 8762 |
| 0.79 | 454 | 935 | 2078 | 4142 | 8814 |
| 0.8 | 448 | 942 | 2116 | 4163 | 8418 |
| 0.81 | 448 | 932 | 1999 | 4053 | 8731 |
| 0.82 | 457 | 923 | 2033 | 4085 | 8469 |
| 0.83 | 447 | 931 | 1984 | 4179 | 8509 |
| 0.84 | 447 | 931 | 1989 | 4202 | 8489 |
| 0.85 | 458 | 941 | 2036 | 4122 | 8486 |
| 0.86 | 471 | 939 | 2045 | 4066 | 8524 |
| 0.87 | 455 | 941 | 2056 | 4216 | 8553 |
| 0.88 | 454 | 923 | 2066 | 4105 | 8590 |
| 0.89 | 448 | 954 | 1995 | 4055 | 8461 |
| 0.9 | 448 | 1031 | 1971 | 4055 | 8669 |
| 0.91 | 446 | 1066 | 1971 | 4212 | 8534 |
| 0.92 | 450 | 1007 | 2044 | 4080 | 8646 |
| 0.93 | 445 | 944 | 2104 | 4078 | 8399 |
| 0.94 | 450 | 932 | 2118 | 4058 | 8360 |
| 0.95 | 452 | 938 | 2167 | 4193 | 8877 |
| 0.96 | 458 | 929 | 2102 | 4058 | 8599 |
| 0.97 | 456 | 925 | 2079 | 4066 | 8540 |
| 0.98 | 468 | 924 | 2061 | 4088 | 8611 |
| 0.99 | 496 | 951 | 2050 | 4194 | 8441 |
| 1 | 479 | 930 | 2056 | 4064 | 8769 |

A graph of different colored lines

Description automatically generated

* Threads: 8

Array sizes: Increased from 1 million to 16 million (using the doubling approach)

Cutoff Ratio: Cutoff value/Size of array – 0.5 to 1 (incrementing 0.1 each time)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cutoff/Array size** | **1 Million** | **2 Million** | **4 Million** | **8 Million** | **16 Million** |
| 0.5 | 699 | 1242 | 1775 | 3465 | 7054 |
| 0.51 | 485 | 966 | 2021 | 4067 | 8433 |
| 0.52 | 450 | 977 | 1945 | 4137 | 8512 |
| 0.53 | 446 | 932 | 1948 | 4093 | 8347 |
| 0.54 | 458 | 940 | 1948 | 4051 | 8526 |
| 0.55 | 476 | 930 | 1962 | 4034 | 8319 |
| 0.56 | 482 | 934 | 2055 | 4125 | 8478 |
| 0.57 | 456 | 939 | 1998 | 4149 | 8358 |
| 0.58 | 463 | 940 | 1973 | 4032 | 8531 |
| 0.59 | 453 | 944 | 1964 | 4046 | 8313 |
| 0.6 | 450 | 933 | 1974 | 4170 | 8501 |
| 0.61 | 452 | 936 | 1946 | 4052 | 8432 |
| 0.62 | 445 | 927 | 1973 | 4073 | 8459 |
| 0.63 | 465 | 944 | 2031 | 4018 | 8470 |
| 0.64 | 442 | 934 | 2061 | 4184 | 8414 |
| 0.65 | 447 | 932 | 1994 | 4044 | 8557 |
| 0.66 | 446 | 957 | 1942 | 4047 | 8429 |
| 0.67 | 446 | 985 | 1943 | 4020 | 8505 |
| 0.68 | 445 | 953 | 1964 | 4187 | 8352 |
| 0.69 | 449 | 955 | 1972 | 4017 | 8499 |
| 0.7 | 473 | 931 | 1976 | 4028 | 8427 |
| 0.71 | 466 | 945 | 1961 | 4060 | 8535 |
| 0.72 | 465 | 928 | 2021 | 4169 | 8347 |
| 0.73 | 458 | 935 | 2031 | 4019 | 8499 |
| 0.74 | 472 | 930 | 1935 | 4076 | 8396 |
| 0.75 | 509 | 944 | 1949 | 4029 | 8527 |
| 0.76 | 467 | 931 | 2000 | 4150 | 8418 |
| 0.77 | 477 | 927 | 1952 | 4038 | 8503 |
| 0.78 | 474 | 951 | 1959 | 4028 | 8471 |
| 0.79 | 466 | 948 | 1940 | 4076 | 8461 |
| 0.8 | 471 | 982 | 2044 | 4168 | 8429 |
| 0.81 | 460 | 956 | 2068 | 4037 | 8425 |
| 0.82 | 491 | 933 | 1965 | 4045 | 8582 |
| 0.83 | 451 | 987 | 1973 | 4028 | 8330 |
| 0.84 | 452 | 970 | 1977 | 4146 | 8513 |
| 0.85 | 460 | 969 | 1987 | 4030 | 8313 |
| 0.86 | 483 | 949 | 1962 | 4017 | 8554 |
| 0.87 | 456 | 942 | 1962 | 4219 | 8348 |
| 0.88 | 456 | 929 | 2068 | 4100 | 8436 |
| 0.89 | 446 | 935 | 2028 | 4090 | 8397 |
| 0.9 | 474 | 994 | 1969 | 4007 | 8498 |
| 0.91 | 450 | 930 | 1953 | 4082 | 8353 |
| 0.92 | 454 | 938 | 1967 | 4441 | 8548 |
| 0.93 | 443 | 927 | 1967 | 4292 | 8326 |
| 0.94 | 455 | 934 | 1975 | 4108 | 8395 |
| 0.95 | 461 | 923 | 1967 | 4178 | 8374 |
| 0.96 | 459 | 924 | 2036 | 4078 | 8444 |
| 0.97 | 466 | 926 | 2034 | 4036 | 8409 |
| 0.98 | 449 | 930 | 1966 | 4023 | 8413 |
| 0.99 | 451 | 925 | 1975 | 4157 | 8412 |
| 1 | 452 | 986 | 1964 | 4187 | 8497 |

A graph of different colored lines

Description automatically generated

1. To verify how many threads give us the most optimum solution, we’ll check different thread sizes from 2 to 16 for an array of size 1000000. The cutoff ratios will be changing from 0.50 to 1.00 with 0.1 increments in all cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cutoff/Array size** | **2 Threads** | **4 Threads** | **8 Threads** | **16 Threads** |
| 0.5 | 805 | 813 | 732 | 680 |
| 0.51 | 459 | 507 | 478 | 689 |
| 0.52 | 437 | 469 | 469 | 539 |
| 0.53 | 460 | 468 | 464 | 468 |
| 0.54 | 451 | 457 | 467 | 465 |
| 0.55 | 441 | 467 | 467 | 465 |
| 0.56 | 442 | 459 | 467 | 468 |
| 0.57 | 450 | 462 | 461 | 462 |
| 0.58 | 454 | 463 | 480 | 468 |
| 0.59 | 464 | 457 | 470 | 465 |
| 0.6 | 451 | 458 | 462 | 483 |
| 0.61 | 455 | 457 | 452 | 485 |
| 0.62 | 440 | 457 | 456 | 463 |
| 0.63 | 463 | 460 | 459 | 457 |
| 0.64 | 432 | 480 | 455 | 465 |
| 0.65 | 430 | 487 | 462 | 466 |
| 0.66 | 441 | 482 | 453 | 468 |
| 0.67 | 450 | 480 | 458 | 464 |
| 0.68 | 432 | 466 | 452 | 493 |
| 0.69 | 437 | 489 | 459 | 472 |
| 0.7 | 434 | 502 | 462 | 463 |
| 0.71 | 433 | 470 | 461 | 476 |
| 0.72 | 428 | 465 | 500 | 464 |
| 0.73 | 445 | 466 | 526 | 458 |
| 0.74 | 430 | 460 | 452 | 470 |
| 0.75 | 433 | 464 | 459 | 463 |
| 0.76 | 433 | 459 | 473 | 460 |
| 0.77 | 445 | 466 | 470 | 462 |
| 0.78 | 433 | 462 | 467 | 470 |
| 0.79 | 431 | 457 | 451 | 470 |
| 0.8 | 425 | 453 | 450 | 485 |
| 0.81 | 427 | 460 | 450 | 494 |
| 0.82 | 433 | 463 | 462 | 490 |
| 0.83 | 427 | 467 | 453 | 483 |
| 0.84 | 438 | 459 | 452 | 467 |
| 0.85 | 437 | 468 | 464 | 490 |
| 0.86 | 429 | 463 | 452 | 461 |
| 0.87 | 441 | 458 | 450 | 464 |
| 0.88 | 427 | 471 | 460 | 457 |
| 0.89 | 433 | 459 | 463 | 465 |
| 0.9 | 427 | 466 | 466 | 461 |
| 0.91 | 436 | 456 | 468 | 457 |
| 0.92 | 434 | 462 | 452 | 454 |
| 0.93 | 436 | 459 | 459 | 460 |
| 0.94 | 441 | 467 | 473 | 462 |
| 0.95 | 454 | 469 | 457 | 463 |
| 0.96 | 441 | 460 | 444 | 464 |
| 0.97 | 436 | 459 | 446 | 454 |
| 0.98 | 453 | 490 | 453 | 460 |
| 0.99 | 468 | 485 | 449 | 462 |
| 1 | 440 | 485 | 456 | 459 |

A graph of thread performance

Description automatically generated

Performing the above steps for an array of size 16 million to observe the outcome.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cutoff/Array size** | **2 Threads** | **4 Threads** | **8 Threads** | **16 Threads** |
| 0.5 | 6980 | 6275 | 6596 | 6821 |
| 0.51 | 8533 | 8198 | 8779 | 8466 |
| 0.52 | 8654 | 8490 | 8919 | 8697 |
| 0.53 | 8756 | 8481 | 9361 | 8504 |
| 0.54 | 8638 | 8271 | 8866 | 8561 |
| 0.55 | 8511 | 8661 | 8910 | 8460 |
| 0.56 | 8595 | 8691 | 9097 | 8622 |
| 0.57 | 8706 | 8774 | 9057 | 8448 |
| 0.58 | 8752 | 8285 | 9159 | 8608 |
| 0.59 | 8895 | 8574 | 9009 | 8558 |
| 0.6 | 8740 | 8477 | 8918 | 8543 |
| 0.61 | 9136 | 8524 | 8806 | 8523 |
| 0.62 | 9260 | 8386 | 9085 | 8373 |
| 0.63 | 9079 | 8433 | 8805 | 9164 |
| 0.64 | 8771 | 8153 | 8974 | 8802 |
| 0.65 | 8833 | 8353 | 8825 | 9114 |
| 0.66 | 8750 | 8345 | 8754 | 8659 |
| 0.67 | 8509 | 8330 | 9265 | 8808 |
| 0.68 | 8694 | 8187 | 8985 | 8978 |
| 0.69 | 8674 | 8387 | 8860 | 9172 |
| 0.7 | 8688 | 8352 | 8562 | 8952 |
| 0.71 | 8623 | 8417 | 8692 | 8802 |
| 0.72 | 8686 | 8252 | 8617 | 9052 |
| 0.73 | 8673 | 8166 | 8693 | 9064 |
| 0.74 | 8717 | 8287 | 8559 | 9131 |
| 0.75 | 8620 | 8110 | 8661 | 8560 |
| 0.76 | 8700 | 8278 | 8767 | 8632 |
| 0.77 | 8940 | 7957 | 8628 | 8514 |
| 0.78 | 8558 | 8196 | 8666 | 8667 |
| 0.79 | 9052 | 7921 | 8654 | 8529 |
| 0.8 | 8913 | 8137 | 8665 | 8465 |
| 0.81 | 8670 | 8324 | 8530 | 8558 |
| 0.82 | 8448 | 8670 | 8737 | 8436 |
| 0.83 | 8676 | 8364 | 8500 | 9132 |
| 0.84 | 8506 | 8468 | 8652 | 8752 |
| 0.85 | 8502 | 8271 | 8582 | 8676 |
| 0.86 | 8496 | 8337 | 8700 | 8471 |
| 0.87 | 8471 | 8294 | 8568 | 8635 |
| 0.88 | 8509 | 8414 | 8697 | 8464 |
| 0.89 | 8381 | 8224 | 8596 | 8533 |
| 0.9 | 8577 | 8542 | 8588 | 8561 |
| 0.91 | 8468 | 8464 | 8635 | 8662 |
| 0.92 | 8532 | 8377 | 8568 | 8513 |
| 0.93 | 8367 | 8184 | 8723 | 8427 |
| 0.94 | 8466 | 8295 | 8762 | 8564 |
| 0.95 | 8402 | 8283 | 8626 | 8445 |
| 0.96 | 8479 | 8357 | 8706 | 8536 |
| 0.97 | 8456 | 8444 | 8643 | 8457 |
| 0.98 | 8468 | 8368 | 8534 | 8715 |
| 0.99 | 8370 | 8308 | 8663 | 8387 |
| 1 | 8636 | 8421 | 8549 | 8522 |

A graph of threads

Description automatically generated

**Code Output Screenshots:**

**Conclusion 1 Snapshots:** Cutoff and Time values for various thread sizes and array sizes

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

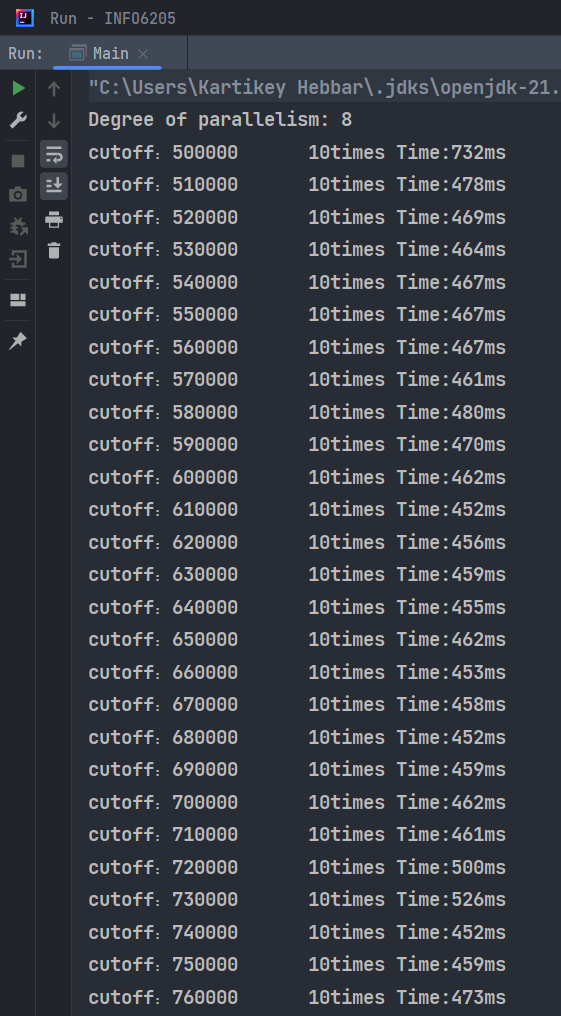
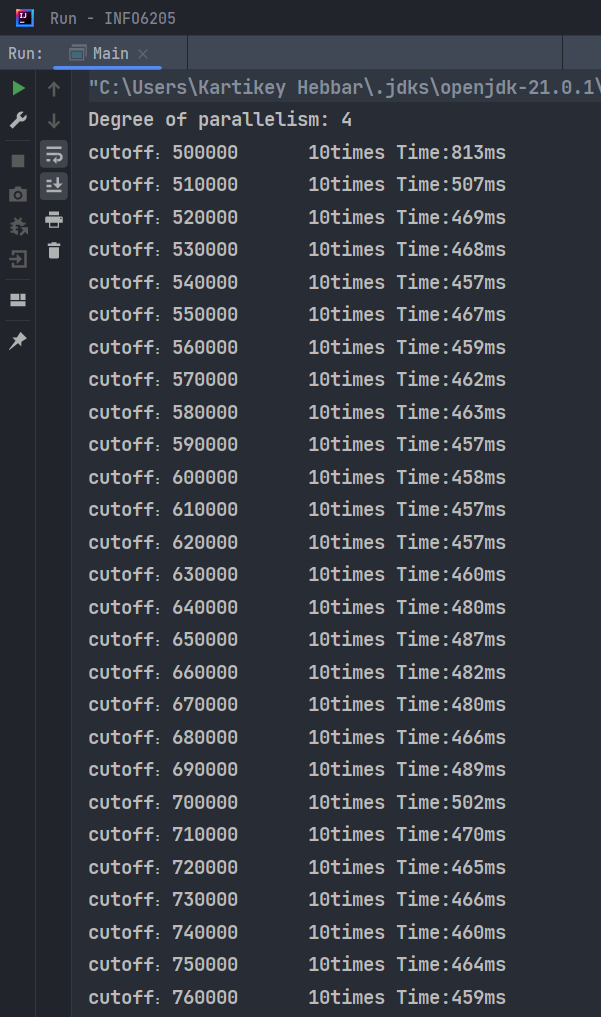
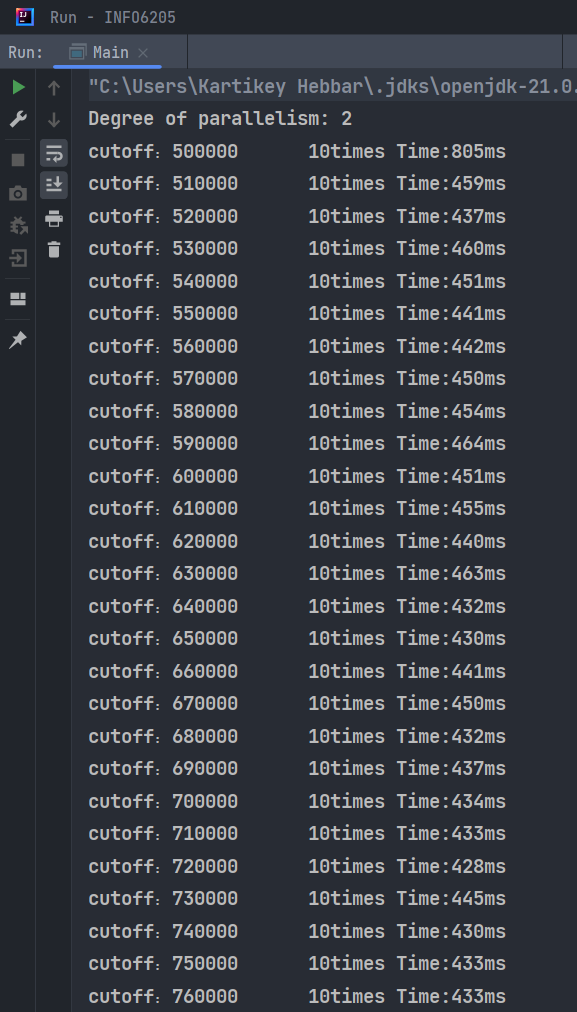
Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Conclusion 2:** Cutoff and time values for various thread sizes when array size is 1 million.



Cutoff and time values for various thread sizes when array size is 16 million.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated