Readme :

Please change the nthreads value to match the number of elements in the array.

I modified the batch file to accept 2 command line arg( after output put $1 $2), so the batch file needs to be run as eg –

**sbatch run\_batch file\_450.txt 450**

( 450 is the number records in the file and file\_450.txt is the file name)

**Q1(a) output**

Tested on array sizes 10,50,150,450 and found results always matching sequential.

login1.stampede(230)$ cat myGPU8560212

Number of elements in the array: 10

 Input Array is: 34 690 440 175 308 690 280 639 528 353

 Sequential CPU: Min number is 34

  Parallel GPU: Min number is 34

login1.stampede(231)$ cat myGPU8560200

Number of elements in the array: 50

 Input Array is: 204 769 527 399 160 743 77 239 71 544 555 679 154 466 70 699 930 698 156 303 548 729 58 118 470 406 982 763 334 454 125 479 811 713 590 135 557 215 204 467 948 403 768 722 226 410 826 894 408 696

 Sequential CPU: Min number is 58

  Parallel GPU: Min number is 58

login1.stampede(247)$ cat myGPU8560303

Number of elements in the array: 150

 Input Array is: 905 736 84 686 229 593 301 520 384 956 618 829 258 67 650 257 725 192 148 309 638 967 300 756 259 718 536 70 804 188 498 47 392 289 550 344 911 400 749 194 415 644 746 107 196 119 919 856 89 728 761 538 843 515 960 203 475 124 984 919 105 261 583 117 105 352 157 538 126 274 541 833 318 760 797 644 210 459 904 325 68 793 692 513 473 66 984 475 536 6 121 403 806 510 100 366 306 326 770 152 556 344 845 127 432 563 527 755 343 114 12 411 433 878 242 633 456 459 328 780 193 672 642 256 908 974 34 146 862 798 584 696 409 539 766 974 278 337 981 419 624 885 14 901 419 359 431 319 381 913

 Sequential CPU: Min number is 6

  Parallel GPU: Min number is 6

login1.stampede(253)$ cat myGPU8560325

Number of elements in the array: 450

 Input Array is: 503 54 64 335 74 163 346 485 904 765 225 763 710 700 325 805 732 804 282 723 475 264 353 335 103 474 341 38 563 590 514 538 654 163 706 162 928 526 33 976 738 78 101 252 877 482 643 752 872 107 821 942 313 410 716 445 313 940 429 686 164 792 515 812 385 597 812 135 268 648 816 717 446 122 530 881 548 288 232 937 381 484 919 456 995 442 826 290 626 113 844 14 447 380 493 448 583 68 492 598 162 424 124 453 123 850 453 51 594 558 704 189 670 31 351 883 513 227 623 798 144 676 202 227 662 59 54 633 368 796 133 207 155 825 146 806 979 474 44 242 124 307 495 549 988 191 322 724 701 490 81 120 597 415 444 496 442 564 254 280 157 489 333 126 276 156 893 54 942 612 890 503 836 768 737 70 854 999 736 915 104 962 563 303 27 559 673 256 98 371 179 930 47 44 835 133 518 374 729 428 652 465 416 242 510 829 674 785 2 424 991 508 589 514 322 835 783 142 823 187 749 832 439 781 904 307 700 652 747 737 888 28 130 847 249 835 213 310 656 10 71 189 814 52 71 504 197 908 91 565 651 843 590 782 94 390 30 289 590 130 857 61 227 453 701 94 973 674 137 268 116 380 559 808 56 460 198 797 963 871 704 867 450 247 641 413 772 388 72 920 819 847 374 846 304 514 872 100 487 939 576 412 832 386 604 543 345 8 360 471 746 580 695 327 621 422 203 240 782 90 2 750 250 52 651 45 38 553 870 858 749 971 237 796 463 733 795 296 568 919 170 256 97 560 466 713 663 766 285 994 526 357 972 130 119 631 945 874 360 197 957 817 79 901 899 746 617 271 591 235 504 494 726 903 576 443 278 221 162 33 308 889 936 925 993 641 853 646 608 946 34 821 983 514 17 669 457 133 990 129 275 613 713 570 416 763 39 18 843 464 269 208 673 557 724 293 609 13 738 194 516 735 556 45 184 712 294 917 114 586 136 581 303 97 675 816 570 328 873 418 23 912 782 851 185 679 652 787 863 475

 Sequential CPU: Min number is 2

  Parallel GPU: Min number is 2

**Q1(b) output**

Tested on array sizes 10,50,150,450 and found results always matching sequential and correct as per the requirements in the problem. Pasting the two sample outputs here.

login4.stampede(65)$ cat myGPU8561568

Number of elements: 10

Input Array is: 34 690 440 175 308 690 280 639 528 353

Output Array :CPU : 4 0 0 5 8 0 0 9 8 3

Output  array: GPU : 4 0 0 5 8 0 0 9 8 3

Number of elements: 50

Input Array is: 204 769 527 399 160 743 77 239 71 544 555 679 154 466 70 699 930 698 156 303 548 729 58 118 470 406 982 763 334 454 125 479 811 713 590 135 557 215 204 467 948 403 768 722 226 410 826 894 408 696

Output Array :CPU : 4 9 7 9 0 3 7 9 1 4 5 9 4 6 0 9 0 8 6 3 8 9 8 8 0 6 2 3 4 4 5 9 1 3 0 5 7 5 4 7 8 3 8 2 6 0 6 4 8 6

Output  array: GPU : 4 9 7 9 0 3 7 9 1 4 5 9 4 6 0 9 0 8 6 3 8 9 8 8 0 6 2 3 4 4 5 9 1 3 0 5 7 5 4 7 8 3 8 2 6 0 6 4 8 6

**Q3 output**

Tested on array sizes 10,50,100,150,450 and found results always matching sequential and correctly sorted. Pasting the two sample outputs here.

login2.stampede(33)$ cat myGPU8562989

Number of elements in the array : 10

Input Array is: 34 690 440 175 308 690 280 639 528 353

Sorted array using CPU(Sequential) sort:

34 175 280 308 353 440 528 639 690 690

Sorted array using GPU(Parallel) sort:

 34 175 280 308 353 440 528 639 690 690

login2.stampede(45)$ cat myGPU8563016

Number of elements in the array : 450

Sorted array using CPU(Sequential) sort:

2 2 8 10 13 14 17 18 23 27 28 30 31 33 33 34 38 38 39 44 44 45 45 47 51 52 52 54 54 54 56 59 61 64 68 70 71 71 72 74 78 79 81 90 91 94 94 97 97 98 100 101 103 104 107 113 114 116 119 120 122 123 124 124 126 129 130 130 130 133 133 133 135 136 137 142 144 146 155 156 157 162 162 162 163 163 164 170 179 184 185 187 189 189 191 194 197 197 198 202 203 207 208 213 221 225 227 227 227 232 235 237 240 242 242 247 249 250 252 254 256 256 264 268 268 269 271 275 276 278 280 282 285 288 289 290 293 294 296 303 303 304 307 307 308 310 313 313 322 322 325 327 328 333 335 335 341 345 346 351 353 357 360 360 368 371 374 374 380 380 381 385 386 388 390 410 412 413 415 416 416 418 422 424 424 428 429 439 442 442 443 444 445 446 447 448 450 453 453 453 456 457 460 463 464 465 466 471 474 474 475 475 482 484 485 487 489 490 492 493 494 495 496 503 503 504 504 508 510 513 514 514 514 514 515 516 518 526 526 530 538 543 548 549 553 556 557 558 559 559 560 563 563 564 565 568 570 570 576 576 580 581 583 586 589 590 590 590 591 594 597 597 598 604 608 609 612 613 617 621 623 626 631 633 641 641 643 646 648 651 651 652 652 652 654 656 662 663 669 670 673 673 674 674 675 676 679 686 695 700 700 701 701 704 704 706 710 712 713 713 716 717 723 724 724 726 729 732 733 735 736 737 737 738 738 746 746 747 749 749 750 752 763 763 765 766 768 772 781 782 782 782 783 785 787 792 795 796 796 797 798 804 805 806 808 812 812 814 816 816 817 819 821 821 823 825 826 829 832 832 835 835 835 836 843 843 844 846 847 847 850 851 853 854 857 858 863 867 870 871 872 872 873 874 877 881 883 888 889 890 893 899 901 903 904 904 908 912 915 917 919 919 920 925 928 930 936 937 939 940 942 942 945 946 957 962 963 971 972 973 976 979 983 988 990 991 993 994 995 999

Sorted array using GPU(Parallel) sort:

 2 2 8 10 13 14 17 18 23 27 28 30 31 33 33 34 38 38 39 44 44 45 45 47 51 52 52 54 54 54 56 59 61 64 68 70 71 71 72 74 78 79 81 90 91 94 94 97 97 98 100 101 103 104 107 113 114 116 119 120 122 123 124 124 126 129 130 130 130 133 133 133 135 136 137 142 144 146 155 156 157 162 162 162 163 163 164 170 179 184 185 187 189 189 191 194 197 197 198 202 203 207 208 213 221 225 227 227 227 232 235 237 240 242 242 247 249 250 252 254 256 256 264 268 268 269 271 275 276 278 280 282 285 288 289 290 293 294 296 303 303 304 307 307 308 310 313 313 322 322 325 327 328 333 335 335 341 345 346 351 353 357 360 360 368 371 374 374 380 380 381 385 386 388 390 410 412 413 415 416 416 418 422 424 424 428 429 439 442 442 443 444 445 446 447 448 450 453 453 453 456 457 460 463 464 465 466 471 474 474 475 475 482 484 485 487 489 490 492 493 494 495 496 503 503 504 504 508 510 513 514 514 514 514 515 516 518 526 526 530 538 543 548 549 553 556 557 558 559 559 560 563 563 564 565 568 570 570 576 576 580 581 583 586 589 590 590 590 591 594 597 597 598 604 608 609 612 613 617 621 623 626 631 633 641 641 643 646 648 651 651 652 652 652 654 656 662 663 669 670 673 673 674 674 675 676 679 686 695 700 700 701 701 704 704 706 710 712 713 713 716 717 723 724 724 726 729 732 733 735 736 737 737 738 738 746 746 747 749 749 750 752 763 763 765 766 768 772 781 782 782 782 783 785 787 792 795 796 796 797 798 804 805 806 808 812 812 814 816 816 817 819 821 821 823 825 826 829 832 832 835 835 835 836 843 843 844 846 847 847 850 851 853 854 857 858 863 867 870 871 872 872 873 874 877 881 883 888 889 890 893 899 901 903 904 904 908 912 915 917 919 919 920 925 928 930 936 937 939 940 942 942 945 946 957 962 963 971 972 973 976 979 983 988 990 991 993 994 995 999