Nina Żukowska 148278, Antoni Solarski 148270 – SI3

EX 1. Testing sorting algorithms, arrays with 100 random elements from [100, 999]

818 297 222 680 243 731 787 443 147 205 136 250 572 285 650 381 627 252 747 574 933 931 691 261
489 362 145 482 201 834 567 100 718 319 295 876 738 985 862 426 648 386 528 231 943 496 488 973
451 882 553 163 396 361 395 480 251 933 187 689 566 154 415 348 503 260 449 616 304 937 322 771
669 565 349 871 286 946 736 140 255 577 415 968 934 272 513 804 612 926 431 635 438 932 761 468
661 182 922 745

element 0 IS: 100; SS: 100; BS: 100; HS: 100; QS: 100; MS: 100; SHS: 100; CS: 100;

element 1 IS: 136; SS: 136; BS: 136; HS: 136; QS: 136; MS: 136; SHS: 136; CS: 136;

element 2 IS: 140; SS: 140; BS: 140; HS: 140; QS: 140; MS: 140; SHS: 140; CS: 140;

element 3 IS: 145; SS: 145; BS: 145; HS: 145; QS: 145; MS: 145; SHS: 145; CS: 145;

element 4 IS: 147; SS: 147; BS: 147; HS: 147; QS: 147; MS: 147; SHS: 147; CS: 147;

element 5 IS: 154; SS: 154; BS: 154; HS: 154; QS: 154; MS: 154; SHS: 154; CS: 154;

element 6 IS: 163; SS: 163; BS: 163; HS: 163; QS: 163; MS: 163; SHS: 163; CS: 163;

element 7 IS: 182; SS: 182; BS: 182; HS: 182; QS: 182; MS: 182; SHS: 182; CS: 182;

element 8 IS: 187; SS: 187; BS: 187; HS: 187; QS: 187; MS: 187; SHS: 187; CS: 187;

element 9 IS: 201; SS: 201; BS: 201; HS: 201; QS: 201; MS: 201; SHS: 201; CS: 201;

element 10 IS: 205; SS: 205; BS: 205; HS: 205; QS: 205; MS: 205; SHS: 205; CS: 205;

element 11 IS: 222; SS: 222; BS: 222; HS: 222; QS: 222; MS: 222; SHS: 222; CS: 222;

element 12 IS: 231; SS: 231; BS: 231; HS: 231; QS: 231; MS: 231; SHS: 231; CS: 231;

element 13 IS: 243; SS: 243; BS: 243; HS: 243; QS: 243; MS: 243; SHS: 243; CS: 243;

element 14 IS: 250; SS: 250; BS: 250; HS: 250; QS: 250; MS: 250; SHS: 250; CS: 250;

element 15 IS: 251; SS: 251; BS: 251; HS: 251; QS: 251; MS: 251; SHS: 251; CS: 251;

element 16 IS: 252; SS: 252; BS: 252; HS: 252; QS: 252; MS: 252; SHS: 252; CS: 252;

element 17 IS: 255; SS: 255; BS: 255; HS: 255; QS: 255; MS: 255; SHS: 255; CS: 255;

element 18 IS: 260; SS: 260; BS: 260; HS: 260; QS: 260; MS: 260; SHS: 260; CS: 260;

element 19 IS: 261; SS: 261; BS: 261; HS: 261; QS: 261; MS: 261; SHS: 261; CS: 261;

element 20 IS: 272; SS: 272; BS: 272; HS: 272; QS: 272; MS: 272; SHS: 272; CS: 272;

element 21 IS: 285; SS: 285; BS: 285; HS: 285; QS: 285; MS: 285; SHS: 285; CS: 285;

element 22 IS: 286; SS: 286; BS: 286; HS: 286; QS: 286; MS: 286; SHS: 286; CS: 286;

element 23 IS: 295; SS: 295; BS: 295; HS: 295; QS: 295; MS: 295; SHS: 295; CS: 295;

element 24 IS: 297; SS: 297; BS: 297; HS: 297; QS: 297; MS: 297; SHS: 297; CS: 297;

element 25 IS: 304; SS: 304; BS: 304; HS: 304; QS: 304; MS: 304; SHS: 304; CS: 304;

element 26 IS: 319; SS: 319; BS: 319; HS: 319; QS: 319; MS: 319; SHS: 319; CS: 319;

element 27 IS: 322; SS: 322; BS: 322; HS: 322; QS: 322; MS: 322; SHS: 322; CS: 322;

element 28	IS: 348; SS: 348; BS: 348; HS: 348; QS: 348; MS: 348; SHS: 348; CS: 348;
element 29	IS: 349; SS: 349; BS: 349; HS: 349; QS: 349; MS: 349; SHS: 349; CS: 349;
element 30	IS: 361; SS: 361; BS: 361; HS: 361; QS: 361; MS: 361; SHS: 361; CS: 361;
element 31	IS: 362; SS: 362; BS: 362; HS: 362; QS: 362; MS: 362; SHS: 362; CS: 362;
element 32	IS: 381; SS: 381; BS: 381; HS: 381; QS: 381; MS: 381; SHS: 381; CS: 381;
element 33	IS: 386; SS: 386; BS: 386; HS: 386; QS: 386; MS: 386; SHS: 386; CS: 386;
element 34	IS: 395; SS: 395; BS: 395; HS: 395; QS: 395; MS: 395; SHS: 395; CS: 395;
element 35	IS: 396; SS: 396; BS: 396; HS: 396; QS: 396; MS: 396; SHS: 396; CS: 396;
element 36	IS: 415; SS: 415; BS: 415; HS: 415; QS: 415; MS: 415; SHS: 415; CS: 415;
element 37	IS: 415; SS: 415; BS: 415; HS: 415; QS: 415; MS: 415; SHS: 415; CS: 415;
element 38	IS: 426; SS: 426; BS: 426; HS: 426; QS: 426; MS: 426; SHS: 426; CS: 426;
element 39	IS: 431; SS: 431; BS: 431; HS: 431; QS: 431; MS: 431; SHS: 431; CS: 431;
element 40	IS: 438; SS: 438; BS: 438; HS: 438; QS: 438; MS: 438; SHS: 438; CS: 438;
element 41	IS: 443; SS: 443; BS: 443; HS: 443; QS: 443; MS: 443; SHS: 443; CS: 443;
element 42	IS: 449; SS: 449; BS: 449; HS: 449; QS: 449; MS: 449; SHS: 449; CS: 449;
element 43	IS: 451; SS: 451; BS: 451; HS: 451; QS: 451; MS: 451; SHS: 451; CS: 451;
element 44	IS: 468; SS: 468; BS: 468; HS: 468; QS: 468; MS: 468; SHS: 468; CS: 468;
element 45	IS: 480; SS: 480; BS: 480; HS: 480; QS: 480; MS: 480; SHS: 480; CS: 480;
element 46	IS: 482; SS: 482; BS: 482; HS: 482; QS: 482; MS: 482; SHS: 482; CS: 482;
element 47	IS: 488; SS: 488; BS: 488; HS: 488; QS: 488; MS: 488; SHS: 488; CS: 488;
element 48	IS: 489; SS: 489; BS: 489; HS: 489; QS: 489; MS: 489; SHS: 489; CS: 489;
element 49	IS: 496; SS: 496; BS: 496; HS: 496; QS: 496; MS: 496; SHS: 496; CS: 496;
element 50	IS: 503; SS: 503; BS: 503; HS: 503; QS: 503; MS: 503; SHS: 503; CS: 503;
element 51	IS: 513; SS: 513; BS: 513; HS: 513; QS: 513; MS: 513; SHS: 513; CS: 513;
element 52	IS: 528; SS: 528; BS: 528; HS: 528; QS: 528; MS: 528; SHS: 528; CS: 528;
element 53	IS: 553; SS: 553; BS: 553; HS: 553; QS: 553; MS: 553; SHS: 553; CS: 553;
element 54	IS: 565; SS: 565; BS: 565; HS: 565; QS: 565; MS: 565; SHS: 565; CS: 565;
element 55	IS: 566; SS: 566; BS: 566; HS: 566; QS: 566; MS: 566; SHS: 566; CS: 566;
element 56	IS: 567; SS: 567; BS: 567; HS: 567; QS: 567; MS: 567; SHS: 567; CS: 567;
element 57	IS: 572; SS: 572; BS: 572; HS: 572; QS: 572; MS: 572; SHS: 572; CS: 572;
element 58	IS: 574; SS: 574; BS: 574; HS: 574; QS: 574; MS: 574; SHS: 574; CS: 574;
element 59	IS: 577; SS: 577; BS: 577; HS: 577; QS: 577; MS: 577; SHS: 577; CS: 577;
element 60	IS: 612; SS: 612; BS: 612; HS: 612; QS: 612; MS: 612; SHS: 612; CS: 612;
element 61	IS: 616; SS: 616; BS: 616; HS: 616; QS: 616; MS: 616; SHS: 616; CS: 616;

element 62	IS: 627; SS: 627; BS: 627; HS: 627; QS: 627; MS: 627; SHS: 627; CS: 627;
element 63	IS: 635; SS: 635; BS: 635; HS: 635; QS: 635; MS: 635; SHS: 635; CS: 635;
element 64	IS: 648; SS: 648; BS: 648; HS: 648; QS: 648; MS: 648; SHS: 648; CS: 648;
element 65	IS: 650; SS: 650; BS: 650; HS: 650; QS: 650; MS: 650; SHS: 650; CS: 650;
element 66	IS: 661; SS: 661; BS: 661; HS: 661; QS: 661; MS: 661; SHS: 661; CS: 661;
element 67	IS: 669; SS: 669; BS: 669; HS: 669; QS: 669; MS: 669; SHS: 669; CS: 669;
element 68	IS: 680; SS: 680; BS: 680; HS: 680; QS: 680; MS: 680; SHS: 680; CS: 680;
element 69	IS: 689; SS: 689; BS: 689; HS: 689; QS: 689; MS: 689; SHS: 689; CS: 689;
element 70	IS: 691; SS: 691; BS: 691; HS: 691; QS: 691; MS: 691; SHS: 691; CS: 691;
element 71	IS: 718; SS: 718; BS: 718; HS: 718; QS: 718; MS: 718; SHS: 718; CS: 718;
element 72	IS: 731; SS: 731; BS: 731; HS: 731; QS: 731; MS: 731; SHS: 731; CS: 731;
element 73	IS: 736; SS: 736; BS: 736; HS: 736; QS: 736; MS: 736; SHS: 736; CS: 736;
element 74	IS: 738; SS: 738; BS: 738; HS: 738; QS: 738; MS: 738; SHS: 738; CS: 738;
element 75	IS: 745; SS: 745; BS: 745; HS: 745; QS: 745; MS: 745; SHS: 745; CS: 745;
element 76	IS: 747; SS: 747; BS: 747; HS: 747; QS: 747; MS: 747; SHS: 747; CS: 747;
element 77	IS: 761; SS: 761; BS: 761; HS: 761; QS: 761; MS: 761; SHS: 761; CS: 761;
element 78	IS: 771; SS: 771; BS: 771; HS: 771; QS: 771; MS: 771; SHS: 771; CS: 771;
element 79	IS: 787; SS: 787; BS: 787; HS: 787; QS: 787; MS: 787; SHS: 787; CS: 787;
element 80	IS: 804; SS: 804; BS: 804; HS: 804; QS: 804; MS: 804; SHS: 804; CS: 804;
element 81	IS: 818; SS: 818; BS: 818; HS: 818; QS: 818; MS: 818; SHS: 818; CS: 818;
element 82	IS: 834; SS: 834; BS: 834; HS: 834; QS: 834; MS: 834; SHS: 834; CS: 834;
element 83	IS: 862; SS: 862; BS: 862; HS: 862; QS: 862; MS: 862; SHS: 862; CS: 862;
element 84	IS: 871; SS: 871; BS: 871; HS: 871; QS: 871; MS: 871; SHS: 871; CS: 871;
element 85	IS: 876; SS: 876; BS: 876; HS: 876; QS: 876; MS: 876; SHS: 876; CS: 876;
element 86	IS: 882; SS: 882; BS: 882; HS: 882; QS: 882; MS: 882; SHS: 882; CS: 882;
element 87	IS: 922; SS: 922; BS: 922; HS: 922; QS: 922; MS: 922; SHS: 922; CS: 922;
element 88	IS: 926; SS: 926; BS: 926; HS: 926; QS: 926; MS: 926; SHS: 926; CS: 926;
element 89	IS: 931; SS: 931; BS: 931; HS: 931; QS: 931; MS: 931; SHS: 931; CS: 931;
element 90	IS: 932; SS: 932; BS: 932; HS: 932; QS: 932; MS: 932; SHS: 932; CS: 932;
element 91	IS: 933; SS: 933; BS: 933; HS: 933; QS: 933; MS: 933; SHS: 933; CS: 933;
element 92	IS: 933; SS: 933; BS: 933; HS: 933; QS: 933; MS: 933; SHS: 933; CS: 933;
element 93	IS: 934; SS: 934; BS: 934; HS: 934; QS: 934; MS: 934; SHS: 934; CS: 934;
element 94	IS: 937; SS: 937; BS: 937; HS: 937; QS: 937; MS: 937; SHS: 937; CS: 937;
element 95	IS: 943; SS: 943; BS: 943; HS: 943; QS: 943; MS: 943; SHS: 943; CS: 943;

element 96 IS: 946; SS: 946; BS: 946; HS: 946; QS: 946; MS: 946; SHS: 946; CS: 946;

element 97 IS: 968; SS: 968; BS: 968; HS: 968; QS: 968; MS: 968; SHS: 968; CS: 968;

element 98 IS: 973; SS: 973; BS: 973; HS: 973; QS: 973; MS: 973; SHS: 973; CS: 973;

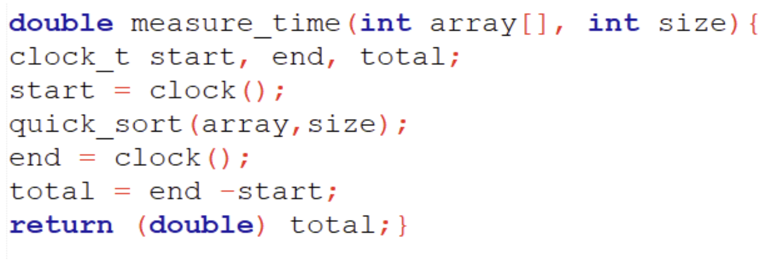
element 99 IS: 985; SS: 985; BS: 985; HS: 985; QS: 985; MS: 985; SHS: 985; CS: 985;

EX 2.

no. of elements\data	A-shaped	V-shaped	normal distribution
49999	2283	62	17
50000	2292	65	17
50001	2237	70	17

Remarks:

1. The code responsible for measuring the cpu time:



```
double measure_time(int array[], int size){
clock_t start, end, total;
start = clock();
quick_sort(array,size);
end = clock();
total = end -start;
return (double) total;}
```

Analysis:

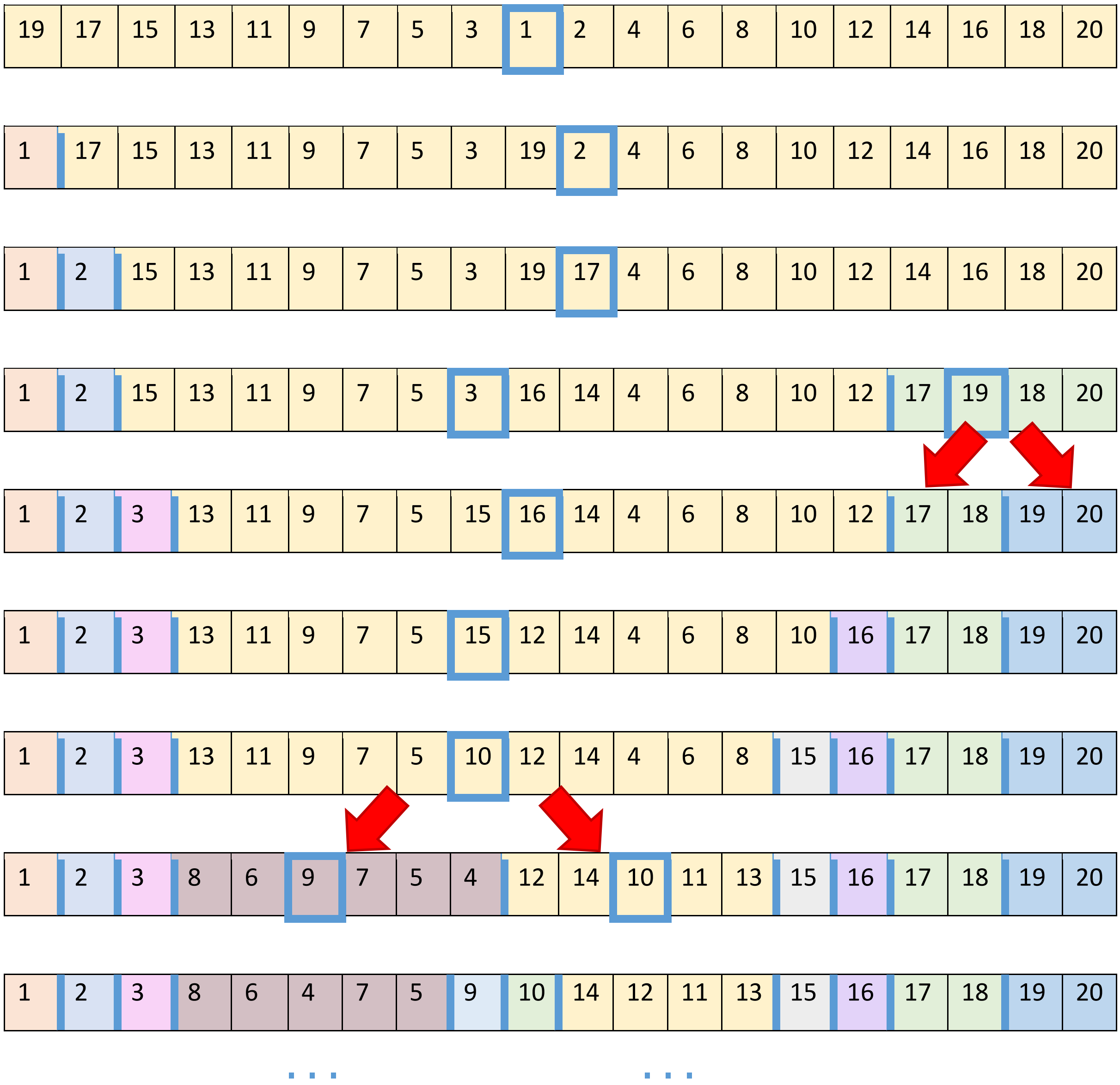
Looking at the data provided for the randomly distributed data, it is apparent that QuickSort(QS) performs a great deal better than in the other cases - one would assume that it would be O(nlog(n))-which indeed is true, that is not because QS always manages to find the ideal pivot, but because the average case is much closer computationally to the best case than to the worst case.

Looking at the V and A-shaped data one can clearly see that QS performs poorly, that is because the middle element of the sorted array is selected to be the pivot, which in both cases is the extremum(minimum and maximum - 1 accordingly - in the first iteration), next it switches place with an element currently occupying its final destination- which in theory is another extremum (if pivot= minimum, then it's the maximum element and the other way around). This works especially well (or poor, the terminology is not objective in this case) with A-shaped arrays with - the scenario presented above is true then. This does not hold for the V-shaped data and that is because even small glitches- (not taking the extremum as the next pivot) reduce the complexity greatly (snowball effect). Although in principle the A- and V- shaped data should produce similar results, they do not - that is because QS stops making bad decisions while working on the array(explained in the "Remarks").

As for the row analysis, nothing too unexpected happens, even though for the A-shaped data QS performs better for 50001 elements it performs better than for 49999 elements, considering such large numbers it isn't saying much, however one can clearly state that for the A-shaped data the overall tendency is up. More surprising might be the fact, that the V-shaped data is consistently performing worse, as shown in the remark section and in the charts of the report, QS on V-shaped data fails to perform in a consistent manner – it is definitely O(n^2), but as stated at some point it fails to choose a "bad pivot", as for QS on randomly distributed data – it performs in a predictable way (worst case not occurring => O(nlogn)). Another interesting remark might be that for some implementations of the A-shaped array generator even-numbered arrays fed as input to QS result in a fat greater running time of QS than arrays od an odd length, - once again this might be due to the process of QS being unstable – same as for the V-shaped data.

REMARK on the workings of the QS on V- shaped data.

In contrast to the A-shaped data the V-shaped (at least in our implementation) data does not consistently result in O(n^2) time complexity when fed to the QS (crucial parts marked):



19	17	15	13	11	9	7	5	3	1	2	4	6	8	10	12	14	16	18	20
1	17	15	13	11	9	7	5	3	19	2	4	6	8	10	12	14	16	18	20
1	2	15	13	11	9	7	5	3	19	17	4	6	8	10	12	14	16	18	20
1	2	15	13	11	9	7	5	3	16	14	4	6	8	10	12	17	19	18	20
1	2	3	13	11	9	7	5	15	16	14	4	6	8	10	12	17	18	19	20
1	2	3	13	11	9	7	5	15	12	14	4	6	8	10	16	17	18	19	20
1	2	3	13	11	9	7	5	10	12	14	4	6	8	15	16	17	18	19	20
1	2	3	8	6	9	7	5	4	12	14	10	11	13	15	16	17	18	19	20
1	2	3	8	6	4	7	5	9	10	14	12	11	13	15	16	17	18	19	20

Such "good" pivot choices might seem rare and one can argue that they are outweighed by the "good pivot" choices – nevertheless even one single "good" at the beginning can largely improve the performance of the algorithm, and most of the times – more of such "good" pivot choices happen.