The descending order data used in Program 2.32 is also worst-case data for both bubble sort versions. This is so as this data causes bubble sort to perform all possible swaps and to make the maximum number of bubble passes. The worst case times for Program 2.9 are given below

n	Repetitions	Total time	Time per sort
0	35038	0.549451	0.000016
10	5804	0.549451	0.000095
20	1664	0.549451	0.000330
30	762	0.549451	0.000721
40	435	0.549451	0.001263
50	280	0.549451	0.001962
60	195	0.549451	0.002818
70	144	0.549451	0.003816
80	110	0.549451	0.004995
90	87	0.549451	0.006316
100	71	0.549451	0.007739
200	18	0.549451	0.030525
300	8	0.549451	0.068681
400	5	0.604396	0.120879
500	3	0.604396	0.201465
600	2	0.549451	0.274725
700	2	0.769231	0.384615
800	2	0.989011	0.494505
900	1	0.604396	0.604396
1000	1	0.769231	0.769231

Times are in seconds

The worst case time for the early terminating version of bubble sort (Program 2.13) are

n	Repetitions	Total time	Time per sort
0	34810	0.549451	0.000016
10	5581	0.549451	0.000098
20	1607	0.549451	0.000342
30	738	0.549451	0.000745
40	421	0.549451	0.001305
50	272	0.549451	0.002020
60	189	0.549451	0.002907
70	140	0.549451	0.003925
80	107	0.549451	0.005135
90	85	0.549451	0.006464
100	69	0.549451	0.007963
200	18	0.549451	0.030525
300	8	0.604396	0.075549
400	5	0.604396	0.120879
500	3	0.604396	0.201465
600	2	0.549451	0.274725
700	2	0.824176	0.412088
800	2	0.989011	0.494505
900	1	0.659341	0.659341
1000	1	0.769231	0.769231

Times are in seconds

Comparing the two sets of times, we see that the overhead involved in keeping track of whether or not a swap has been made and then checking for this causes the early terminating version to run slower for small *n*. For large *n*, this overhead does not affect the run time in a measurable way.