The experiments listed below affirm the fault tolerance property of the partial order sorting algorithm.

Experiment 1:

Rationale: Derive the borderline m that gives back the total order for k = n.

For different values of n (300 >= n >= 20), the number of partial orders is fixed at n (k = n) and the partial order size is fixed at

(a) m = 50% of n   20 <= m <= 100
(b) m = 25% of n   100 < m <= 200
(c) m = 20% of n   200 < m <= 300

The following table gives the flag value deciding whether we get back the total order depending on the partial order set size as listed above.

Flag = 0   The total order is obtained from partial orders is correct.
Flag = 1   The total order is obtained from partial orders is incorrect.

Table 1.

n	k	m (derived as % of n)	m (derived from experiment 4)	% (m % of n)	Flag
20	20	10	9	50	1
30	30	15	11	50	1
40	40	20	14	50	1
50	50	25	17	50	1
60	60	30	19	50	1
70	70	35	20	50	1
80	80	40	22	50	1
90	90	45	25	50	1
100	100	50	24	50	1
110	110	27	26	25	1
120	120	30	28	25	1
130	130	32	29	25	1
140	140	35	30	25	1
150	150	37	30	25	1
160	160	40	32	25	1
170	170	42	32	25	1
180	180	45	35	25	1
190	190	47	36	25	1
200	200	50	36	25	1
210	210	42	38	20	1

220	220	44	40	20	1
230	230	46	38	20	1
240	240	48	41	20	1
250	250	50	41	20	1
260	260	52	42	20	1
270	270	54	42	20	1
280	280	56	43	20	1
290	290	58	45	20	1
300	300	60	45	20	1

Observations:

1. The value of m as derived from Experiment 4 in Algorithm Analysis is listed in the table above. The values of m as derived from percentages listed above are compared to those from Experiment 4 to arrive at the borderline m that would get back the total order.

2. Introducing error at these borderline cases of m would give a more accurate outlook on the fault tolerance of the algorithm.

Experiment 2:

Rationale: Derive the affect of errors for the case of borderline m derived from Experiment 1 above that gives back the total order for k = n.

For different values of n (300 >= n >= 20), errors are introduced into partial orders. Here:

p = % of error introduced.
i.e p % of partial orders have single error in them.

The following table gives the flag value deciding whether we get back the total order depending on the partial order set size as listed above.

Flag = 0 The total order is obtained from partial orders is correct.
Flag = 1 The total order is obtained from partial orders is incorrect.

Table 2.1

n = k	m (derived as % of n)	Flag for p = 5% error	Flag for p = 10% error	Flag for p = 20% error	Flag for p = 50% error	Flag for p = 75% error	Flag for p = 90% error	Flag for p = 100% error
20	10	0	0	1	1	0	0	0
30	15	1	1	1	1	0	1	0
40	20	1	1	1	1	1	1	1
50	25	1	1	1	1	1	1	1

60	30	1	1	1	1	1	1	1
70	35	1	1	1	1	1	1	1
80	40	1	1	1	1	1	1	1
90	45	1	1	1	1	1	1	1
100	50	1	1	1	1	1	1	1
110	27	1	1	1	0	0	0	0
120	30	1	1	1	0	0	0	0
130	32	1	1	0	1	0	1	0
140	35	1	1	1	1	1	0	1
150	37	1	1	1	1	0	0	1
160	40	1	1	1	1	1	1	1
170	42	1	1	1	1	1	1	1
180	45	1	1	0	1	1	1	1
190	47	1	1	1	1	1	1	1
200	50	1	1	1	1	1	1	1
210	42	0	0	1	0	0	1	0
220	44	1	0	0	1	0	0	1
230	46	1	1	1	0	1	0	1
240	48	1	1	1	1	0	0	1
250	50	1	1	1	1	0	1	0
260	52	1	1	0	1	1	1	1
270	54	1	1	1	1	1	0	1
280	56	1	1	1	1	1	1	1
290	58	1	1	1	1	1	0	1
300	60	1	1	1	1	1	1	1

Table 2.1

	Number of ‘n’ values giving errors (Total no. of values = 30)	**% error in 30 values of n**
Flag for p = 5% error	2	0.6
Flag for p = 10% error	3	0.9
Flag for p = 20% error	4	1.2
Flag for p = 50% error	4	1.2
Flag for p = 75% error	10	3

Flag for p = 90% error	10	3
Flag for p = 100% error	7	2.1

Observations:

1. A small percentage of error (p <= 10%) introduced cannot deduce the fault tolerance property of the algorithm.

2. As the error introduced is a two-element error, there is a good possibility of getting back the correct total order even if the error is large.

3. Even a 100% two-element error introduced had no greater than 5% chance of the total order being incorrect.