2.1 Elementary Sorts Exercises

Vincent La

June 29, 2017

2.1.1 Show, in the style of the example trace with Algorithm 2.1, how selection sort sorts the array E A S Y Q U E S T I O N.

i	min	0	1	2	3	4	5	6	7	8	9	10	11
		E	A	S	Y	Q	U	E	S	T	I	O	N
0	1	E	A	S	Y	Q	U	E	S	T	I	O	N
1	1	A	E	S	Y	Q	U	E	S	T	I	O	N
2	6	A	E	S	Y	Q	U	E	S	T	I	O	N
3	9	A	E	E	Y	Q	U	S	S	T	I	O	N
4	11	A	E	E	I	Q	U	S	S	T	Y	O	N
5	10	A	E	E	I	N	U	S	S	T	Y	O	Q
6	11	A	E	E	I	N	O	S	S	T	Y	U	Q
7	7	A	E	E	I	N	O	Q	S	T	Y	U	S
8	11	A	E	E	I	N	O	Q	S	T	Y	U	S
9	11	A	E	E	I	N	O	Q	S	S	Y	U	T
10	10	A	E	E	I	N	O	Q	S	S	T	U	Y
		A	E	E	I	N	O	Q	S	S	T	U	Y

2.1.2 The maximum number of exchanges involving a specific item is N exchanges. For example, take a list that is already sorted and then add to the beginning an item that is greater in value than the rest of the list. Specifically, consider the list Y A B C D.

i	min	0	1	2	3	4
		Y	A	B	C	D
0	1	Y	A	B	C	D
1	2	A	Y	B	C	D
2	3	A	B	Y	C	D
3	4	A	B	C	Y	D
4	4	A	B	C	D	Y
		A	B	C	D	Y

Here, we can see Y was exchanged N times. Furthermore, because a selection sort performs one exchange for every array index, then there are at most N exchanges for an array of length N.

On the other hand, because there are N items and N exchanges then on average each item gets exchanged once. (Is this right)

2.1.3 Give an example of an array of N items that maximizes the number of times the test a[j] < a[min] succeeds (and, therefore, min gets updated) during the operation of selection sort.

Answer Hypothetically, the most that a[j] < a[min] can succeed is every time it is called. In fact, a list in descending order does just that. For example, consider the list E D C B A. Before the first exchange, the following compares are made:

(a) D < E, so $min = 1$

(b) C < D, so $min = 2$

(c) B < C, so $min = 3$

(d) A < B, so $min = 4$

(Expand on this answer later)