Names:

Constantino, Michael Jay C.
Vergara, Kimly John A.

Course/Year: BSSE - 2

SW # 2 ANSWERS

1. Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		P	A	U	L	A	N	D	R	E	I	A	R	D	I	E	N	T	E
0	1	P	A	U	L	A	N	D	R	E	I	A	R	D	I	E	N	T	E
1	4	A	P	U	L	A	N	D	R	E	I	A	R	D	I	E	N	T	E
2	10	A	A	U	L	P	N	D	R	E	I	A	R	D	I	E	N	T	E
3	6	A	A	A	L	P	N	D	R	E	I	U	R	D	I	E	N	T	E
4	12	A	A	A	D	P	N	L	R	E	I	U	R	D	I	E	N	T	E
5	8	A	A	A	D	D	N	L	R	E	I	U	R	P	I	E	N	T	E
6	14	A	A	A	D	D	E	L	R	N	I	U	R	P	I	E	N	T	E
7	17	A	A	A	D	D	E	E	R	N	I	U	R	P	I	L	N	T	E
8	9	A	A	A	D	D	E	E	E	N	I	U	R	P	I	L	N	T	R
9	13	A	A	A	D	D	E	E	E	I	N	U	R	P	I	L	N	T	R
10	14	A	A	A	D	D	E	E	E	I	I	U	R	P	N	L	N	T	R
11	13	A	A	A	D	D	E	E	E	I	I	L	R	P	N	U	N	T	R
12	15	A	A	A	D	D	E	E	E	I	I	L	N	P	R	U	N	T	R
13	15	A	A	A	D	D	E	E	E	I	I	L	N	N	R	U	P	T	R
14	15	A	A	A	D	D	E	E	E	I	I	L	N	N	P	U	R	T	R
15	17	A	A	A	D	D	E	E	E	I	I	L	N	N	P	R	U	T	R
16	16	A	A	A	D	D	E	E	E	I	I	L	N	N	P	R	R	T	U
17	17	A	A	A	D	D	E	E	E	I	I	L	N	N	P	R	R	T	U
		A	**A**	**A**	**D**	**D**	**E**	**E**	**E**	**I**	**I**	**L**	**N**	**N**	**P**	**R**	**R**	**T**	**U**

2. Insertion Sort

i	min	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		K	I	M	L	Y	J	O	H	N	V	E	R	G	A	R	A
0	0	K	I	M	L	Y	J	O	H	N	V	E	R	G	A	R	A
1	0	I	K	M	L	Y	J	O	H	N	V	E	R	G	A	R	A
2	2	I	K	M	L	Y	J	O	H	N	V	E	R	G	A	R	A
3	2	I	K	L	M	Y	J	O	H	N	V	E	R	G	A	R	A
4	4	I	K	L	M	Y	J	O	H	N	V	E	R	G	A	R	A
5	1	I	J	K	L	M	Y	O	H	N	V	E	R	G	A	R	A
6	5	I	J	K	L	M	O	Y	H	N	V	E	R	G	A	R	A
7	0	H	I	J	K	L	M	O	Y	N	V	E	R	G	A	R	A
8	6	H	I	J	K	L	M	N	O	Y	V	E	R	G	A	R	A
9	8	H	I	J	K	L	M	N	O	V	Y	E	R	G	A	R	A
10	0	E	H	I	J	K	L	M	N	O	V	Y	R	G	A	R	A
11	9	E	H	I	J	K	L	M	N	O	R	V	Y	G	A	R	A
12	1	E	G	H	I	J	K	L	M	N	O	R	V	Y	A	R	A
13	0	A	E	G	H	I	J	K	L	M	N	O	R	V	Y	R	A
14	12	A	E	G	H	I	J	K	L	M	N	O	R	R	V	Y	A
15	1	A	A	E	G	H	I	J	K	L	M	N	O	R	R	V	Y
		A	**A**	**E**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**R**	**R**	**V**	**Y**

3.

When using Insertion Sort, the best case time complexity is O(n). Insertion Sort performs very efficiently when the array is already sorted or when all keys are identical. In this case, each element is compared with the previous one, and since they are all identical, no swaps are needed. The algorithm simply iterates through the array once, making it O(n).

While, if using Selection Sort, the time complexity is O(n^2). Selection Sort always performs O(n^2) comparisons, regardless of the initial order of the array. For each element, it scans the remaining elements to find the minimum, even if all elements are identical. This results in n*(n-1)/2 comparisons and n swaps, making it less efficient than Insertion Sort in this scenario.

Therefore, Insertion Sort is faster than Selection Sort when processing an array where all the keys are identical.

4.

The most efficient sorting method for this situation is Selection Sort. When using Selection Sort, you scan each crate to determine its shipping time. Since the warehouse can only hold one crate at a time, Selection Sort is more efficient and less expensive. This is because you only move two crates per iteration. In contrast, using Insertion Sort would require moving many crates, making it less efficient.