**QUESTION 1**

1. The array [7, 2,  8,  12, 0, 3, 8, 3]   was sorted and it is now [0, 2, 3, 3, 7, 8, 8, 12]. Was it sorted by a stable sort?

|  |  |  |
| --- | --- | --- |
|  | A. | No |
|  | B. | Yes |
|  | C. | Not enough information to answer |
|  | D. | Being stable or not is not a property of sorting algorithms. |

**10 points**

**QUESTION 2**

1. Assume that the array of tuples [{7,**4**},  {2,**1**},  {8,**0**},  {12,**1**},  {0,**5**},  {3,**4**},  {8,**6**},  {3,**1**}] will be sorted based on the **second number**(written in **bold** font) of the tuple, in **increasing order**, with a **stable** sorting algorithm. Fill in the sorted array bellow. (Fill in a tuple in each box.)

[ {  },{  },{  },{  },{  },{  },{  },{  } ]

**10 points**

**QUESTION 3**

1. Assume the array [7,2,8,12,0,3,8,3] is sorted with the Insertion Sort algorithm discussed in class (the main one, not a variation of it).

Give the array after the first iteration of the outer loop. Separate the values by commas.



Give the array after the second iteration of the outer loop. Separate the values by commas.



**10 points**

**QUESTION 4**

1. Assume the array A = [5,6,7,1,2] is sorted with Selection sort.

Give the array after the **first** iteration of the outer loop. Separate the elements only by commas (no spaces, no brackets).

[  ]

Give the array after the **second** iteration of the outer loop. Separate the elements only by commas (no spaces, no brackets).

[  ]

**10 points**

**QUESTION 5**

1. **Selection sort** is about to process the element at index 4 (in the outer loop). Which part of the array is needed (or may be visited) in this iteration?

|  |  |  |
| --- | --- | --- |
|  |  | Elements at indexes 0 to 4. |
|  |  | Elements at indexes 4 to the end. |
|  |  | All array elements. |
|  |  | Only the element at index 4. |

**10 points**

**QUESTION 6**

1. What algorithm is **stable**?

|  |  |  |
| --- | --- | --- |
|  |  | Selection sort |
|  |  | Insertion sort |
|  |  | Both of the above. |
|  |  | None of the above. |

**10 points**

**QUESTION 7**

1. What algorithm is adaptive?

|  |  |  |
| --- | --- | --- |
|  |  | Selection sort |
|  |  | Insertion sort |
|  |  | Both of the above. |
|  |  | None of the above. |

**10 points**

**QUESTION 8**

1. Which sorting algorithm performs FEWER DATA MOVES for the **worst** case behavior?

|  |  |  |
| --- | --- | --- |
|  |  | Selection sort |
|  |  | Insertion sort |
|  |  | They both have the same order (Theta) of data moves. |
|  |  | It does not make sense to talk about data moves in this contect. |

**10 points**

**QUESTION 9**

1. Assume indirect sorting was used to sort the Data array below in increasing order. (The animals names are the contents of the array. The left number is the index.)

0 panda

1 cat

2 elephant

3 owl

4 duck

5 mouse

6 cheetah

7 lemur

Give the array A of indexes produced by the indirect sorting algorithm.  List only the elements of A separated by commas (no space, no brackets).

A = [  ]

**10 points**

**QUESTION 10**

1. You search for ***fish*** in the sorted array below. When the middle index is computed you can assume rounding down.

0 cat

1 cheetah

2 duck

3 elephant

4 lemur

5 mouse

6 panda

7 owl

List the **middle indexes** produced and used in this search. List them separated by commas (no space, no brackets): 

Give the last values for the left and right indexes (when the loop has finished).

left = 

right = 