Lab W1D5

Question 1. Goal of this question is to understand inversions.

In Slide 13 (Lesson 4), you have been introduced to inversions. In fact it has the following example.

Example. The array arr = {34,8,64,51,32,21} has nine inversions:

(34,8),(34,32),(34,21),(64,51),(64,32),

(64,21),(51,32),(51,21),(32,21).

Apply Bubble Sort on array arr. List all inversions and the number of inversions **after each iteration of the outer loop. (Please complete the table. Add/delete rows as required.)**

Iteration	inversions	# inversions
0	(34,8), (34,32), (34,21), (64,51), (64,32), (64,21), (51,32), (51,21), (32,21).	9

Total number of inversions for Bubble Sort =

Apply Selection Sort on array arr. List all inversions and the number of inversions **after each iteration of the outer loop. (Please complete the table. Add/delete rows as required.)**

Iteration	inversions	# inversions
0	(34,8), (34,32), (34,21), (64,51), (64,32), (64,21), (51,32), (51,21), (32,21).	9

Total number of inversions for Selection Sort =

Apply Insertion Sort on array arr. List all inversions and the number of inversions **after each iteration of the outer loop. (Please complete the table. Add/delete rows as required.)**

Iteration	inversions	# inversions
0	(34,8), (34,32), (34,21), (64,51), (64,32), (64,21), (51,32), (51,21), (32,21).	9

Total number of inversions for Insertion Sort =

Question 2. Aim of this question is to understand amortized cost analysis.

(a) Show all the calculations:

Sample Instance 4 of Clearable Table

add, add, add, add, clear, add, clear add, clear, add, add, add, clear, add, add, clear, add, add, clear, add, add, clear.

(b) Show all the calculations:

Sample Instance 3 of ArrayList with size doubling strategy

A resize just happened from size 16 to size 32.

Question 3. Aim of this question is to better understand amortized cost analysis.

Data structure : ArrayList with size tripling strategy.

Answer all questions below giving detailed explanation.

(a) What is the actual cost of add?
(b) What is the actual cost of resize?
(c) Using traditional worst-case analysis, show that the average cost of an operation is **NOT** constant time.

(d) Consider a sample instance (hint : resize just happened and current size of the array is 9. **(You should never consider current size = 1 for this calculation).** You are adding. Then you resized again)
   a. What is the Amortized_Cost(add)?
   b. What is the Amortized_Cost(resize)?
   c. Through amortized cost analysis show if there is sequence of n operations (some add, some resize) the average cost of an operation is constant time.