Lab W1D5

Question 1.

Bubble Sort

|  |  |  |
| --- | --- | --- |
| Iteration | Inversions | #Inversions |
| 0 | (34, 8), (34,32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 9 |
| 1 | (34, 32), (34, 21), (51, 32), (51, 21), (32, 21) | 5 |
| 2 | (34, 32), (34, 21), (32, 21) | 3 |
| 3 | (32, 21) | 1 |
| 4 |  | 0 |
| 5 |  | 0 |

Total number of inversions for Bubble Sort = 18

Selection Sort

|  |  |  |
| --- | --- | --- |
| Iteration | Inversions | #Inversions |
| 0 | (34, 8), (34,32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 9 |
| 1 | (34, 32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 8 |
| 2 | (64, 51), (64, 32), (64, 34), (51, 32), (51, 34) | 5 |
| 3 | (51, 34), (64, 34) | 2 |
| 4 | (64, 51) | 1 |
| 5 |  | 0 |

Total number of inversions for Selection Sort = 25

Insertion Sort

|  |  |  |
| --- | --- | --- |
| Iteration | Inversions | #Inversions |
| 0 | (34, 8), (34,32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 9 |
| 1 | (34, 32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 8 |
| 2 | (34, 32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 8 |
| 3 | (34, 32), (34, 21), (64, 51), (64, 32), (64, 21), (51, 32), (51, 21), (32, 21) | 8 |
| 4 | (34, 32), (34, 21), (51, 32), (51, 21), (64, 32), (64, 21), (32, 21) | 7 |
| 5 | (32, 21), (34, 21), (51, 21), (64, 21) | 4 |

Total number of inversions for Selection Sort = 44

Question 2.

Aim of this question is to understand amortized cost analysis.

Consider the following sequence of operations:

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | a | a | a | c | a | a | a | a | a | a | c | a | a | a | a | c | a | a | a | a | a | a | a | a | c | #Total |
| 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 | 44(Actual) |
| 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 44(Amortized) |

Actual costs

c(add) = 1 (= 1 cyberdollar)

c(clear) = k (= k cyberdollars), where k is # elements currently in array

Amortized costs

cˆ(add) = 2 (= 2 cyberdollars)

cˆ(clear) = 0 (= 0 cyberdollars)

(a) The actual total cost= 44 cyberdollars

(b) The average actual cost = 44/26 = 22/13 = 1.69 = O(1)

(c) The amortized total cost = 44 cyberdollars

(d) The average amortized cost = 44/26 = 22/13 = 1.69 = O(1)