**Lab W1D5**

**Question 1 (a)**

**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.

1,1,1,1,4,1,1,1,1,1,1,1,3,1,1,2,1,1,2,1,1,2

Total cost = 30

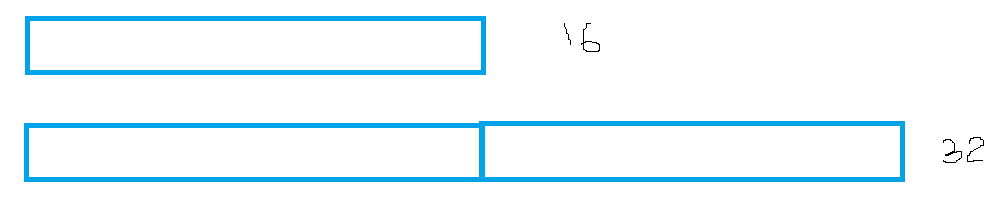
Number of operations = 22

Average cost per operation = 30/22 <= 2

**Question 1 (b)**

**Sample Instance 3 of ArrayList with size doubling strategy**

**A resize just happened from size 16 to size 32.**



Now can add 16 items

Cost of 16 add = 16

Cost resize = 3k = 3 x 32

Total cost = 16 + 96 = 112

Amortize total cost among all adds = 112/ 16 = 7

**Question 2 (a)**

****

Now can add 6 items

Cost of 6 add = 6

Cost resize = 4k = 4 x 9

Total cost = 6 + 36 = 42

Amortize total cost among all adds = 42/ 6 = 7

1. The actual cost of add is between O(1) and O(n).

(b) The actual cost of resize is O(n).

1. Using traditional worst-case analysis, The average cost of an operation is NOT constant time. The worst-case of add operation is O(n) if it is required resizing. So, the average cost of an operation is NOT constant time

**Question 2 (b)**

1. The Amortized\_Cost(add) is 7.

(b) The Amortized\_Cost(resize)is 0.

1. The amortized cost of n operations is O(n). The cost of per operation is O(n)/ n = O(1). Thus, 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.