CS-261: Assignment 2 Written Analysis

Question1: The calculation can be shown in the following table (when = 40):

a)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Append # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Write Cost | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Copy Cost | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total Cumulative  Cost | 1 | 2 | 3 | 4 | 5 | 11 | 12 | 13 | 14 ed11114 | 15 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Append # | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| Write Cost | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Copy Cost | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total Cumulative  Cost | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 |

*b)* As (i.e., the number of appends) grows large, under this strategy for resizing, the average big-O complexity for an append can be derived using the following calculation:

Mathematically, total cost = total write cost ( ) + total copy cost ().

The total write cost, .

Recall (from the table you prepared in part (a)) the copy cost is a sequence as follows: 5, 10, 20, … (a geometric sequence)

Now, what is the cost of the last resize ( term of the sequence)?

The term is derived from the following equation: (where is the first term and is the ratio). For the sequence in our example above:

Now, what is the sum of terms of the above sequence? You can derive that from the following formula:

For the sequence in the example above:

The above equation should work for all integers, , such that ≥ 1. In this instance, represents the resize term number (the 1st resize term, the 2nd resize term, etc.).

However, we are concerned with finding total cost in terms of , so you must express the sum of the resize operations in terms of

You already have determined the resize cost () before. And, the resize cost () is also equal to because the array capacity will double to achieve a size of and there would be half that number of existing elements to copy to the new array. Putting the two together, **you will find out the value of**  **in terms of** .

[Some of you may have noticed that when discussing the total write cost, which is the size of the array, we say that it's N. And now, when discussing , we say that , equating with the capacity. As increases, the ratio of capacity to the size varies between 1 and 2 - just before a resize, they're the same, and just after a resize the capacity is twice the size. is the same for any values between resizes (between terms of the geometric sequence), so we could assign to any value from to . Since the difference is a constant factor, it doesn't affect the big-O, and so we choose because it makes the algebra come out more neatly.]

NOTE: log(2) goes to 1 because we are log base 2

Now, use the and sum equation above () to derive the total copy cost for the given appends:

NOTE: log(2) goes to 1 because we are log base 2

Recall that total cost = total write cost () + total copy cost ().

As we have conducted appends, the average cost can be derived from:

This gives us an amortized complexity of .

Question2:

a) The calculation can be shown in the following table(when N = 40):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Append # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Write Cost | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Copy Cost | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 7 | 0 | d7  9 | 0 | 11 | 0 | 13 | 0 | 15 | 0 | 17 | 0 | 19 |
| Total Cumulative  Cost | 1 | 2 | 3 | 4 | 5 | 11 | 12 | 20 | 21 | 31 | 32 | 44 | 45 | 59 | 60 | 76 | 77 | 95 | 96 | 116 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Append # | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| Write Cost | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Copy Cost | 0 | 21 | 0 | 23 | 0 | 25 | 0 | 27 | 0 | 29 | 0 | 31 | 0 | 33 | 0 | 35 | 0 | 37 | 0 | 39 |
| Total Cumulative  Cost | 117 | 139 | 140 | 164 | 165 | 191 | 192 | 228 | 229 | 259 | 260 | 291 | 292 | 326 | 327 | 363 | 364 | 402 | 403 | 443 |

CS-261: Assignment 2 Written Analysis

b)As (i.e., the number of appends) grows large, under this strategy for resizing, the average big-O complexity for an append can be derived using the following calculation:

Mathematically, total cost = total write cost () + total copy cost ().

The total write cost,  .

Recall (from the table you prepared in part (a)) the copy cost is a sequence as follows: 5, 7, 9, 11, … (an arithmetic sequence)

Now, what is the cost of the last resize ( term of the sequence)?

The term is derived from the following equation, and you will assume the sequence contains k terms: = where is the first term of the sequence and is the difference. In our example above:

Now, what is the sum of terms of the sequence? You can derive that from the following equation:

In our example above:

The above equation should work for all integers, , such that ≥ 1. In this instance, represents the resize term number.

However, we are concerned with finding total cost in terms of , so you must express the sum of the resize operations in terms of

You already have determined the th resize cost () before. Now you determine the *th* resize cost (ak) in terms of (For example, in part (a) when you had = 6, the last resize cost was 5 (holds for all , > 3)). You can consider either of the cases (i.e., is odd or is even 🡪 ). **You will find out the value of**  **in terms of** .

Now, use the and sum equation above () to derive the total copy cost for the given appends:

Recall that total cost = total write cost () + total copy cost () .

As we have conducted appends, the average cost can be derived from:

This gives us an amortized runtime complexity of .