CS-261: Assignment 2 Written Analysis

Question1: The calculation can be shown in the following table (when N = 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	10	0	0	0	0	0	0	0	0	0
Total Cumulative Cost	1	2	3	4	5	11	12	13	14	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 N (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 (S_w) + total copy cost (S_N).

The total write cost, $S_w = N$.

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 (k^{th} term of the sequence)?

The k^{th} term is derived from the following equation: $a_k = a_1 * r^{k-1}$ (where a_1 is the first term and r is the ratio). For the sequence in our example above:

$a_k = 5 * 2^{k-1}$

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

$$S_k = a_1 \left(\frac{1 - r^k}{1 - r}\right)$$

For the sequence in the example above: $S_k = 5\left(\frac{1-2^k}{-1}\right)$

The above equation should work for all integers, k, such that $k \geq 1$. In this instance, k 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 N, so you must express the sum of the k resize operations in terms of N.

You already have determined the k^{th}resize cost (a_k) before. And, the k^{th} resize cost (a_k) is also equal to $\frac{N}{2}$ because the array capacity will double to achieve a size of N 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 k in terms of N**.

[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 S_N, we say that $a_k = \frac{N}{2}$, equating N with the capacity. As N 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. S_N is the same for any values between resizes (between terms of the geometric sequence), so we could assign to a_k any value from $\frac{N}{2}$ to N. Since the difference is a constant factor, it doesn't affect the big-O, and so we choose $\frac{N}{2}$ because it makes the algebra come out more neatly.]

$$a_k = 5 * 2^{k-1} = \frac{N}{2} \rightarrow k = \frac{\log\left(\frac{N}{5}\right)}{\log(2)} = \log\left(\frac{N}{5}\right), for\ N > 0$$

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

$$S_k = 5\left(\frac{1-2^k}{-1}\right) \rightarrow Substitute\ k\ in\ terms\ of\ N$$

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

$$S_N = 5\left(\frac{1-2^{\log\left(\frac{N}{5}\right)}}{-1}\right) = 5\left(\frac{1-\frac{N}{5}}{-1}\right)$$

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

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

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

$$\frac{\{S_w + S_N\}}{N} = \frac{N + 5\left(\frac{1-\frac{N}{5}}{-1}\right)}{N} = \frac{N}{N} + \frac{5\left(-1+\frac{N}{5}\right)}{N} = 1 + \frac{-5+N}{N} = \frac{N}{N} = 1$$

This gives us an amortized complexity of $O(1)$.

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	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 N (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 (S_W) + total copy cost (S_N).

The total write cost, $S_w = N$.

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 (k^{th} term of the sequence)?

The k^{th} term is derived from the following equation, and you will assume the sequence contains k terms: $a_k = a_1 + d(k-1)$ where a_1 is the first term of the sequence and d is the difference. In our example above:

$a_k = 5 + 2(k-1)$

Now, what is the sum of k terms of the sequence? You can derive that from the following equation: $S_k = k\left(\frac{a_1 + a_k}{2}\right)$

In our example above:

$S_k \;= k(\frac{10+2(k-1)}{2})$

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

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

You already have determined the k^{th} resize cost (a_k) before. Now you determine the k^{th} resize cost (a_k) in terms of N (For example, in part (a) when you had $N = 6$, the last resize cost was 5 (holds for all N, $N > 3$)). You can consider either of the cases (i.e., N is odd or N is even → $a_k = 2N$). **You will find out the value of k in terms of N.**

$$a_k = 2N = 5 + 2(k-1) \rightarrow k = N - \frac{3}{2}$$

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

Nappends: $S_N \;= \left(N - \frac{3}{2}\right)\left(\frac{10+2\left(\left(N-\frac{3}{2}\right)-1\right)}{2}\right) = N^2 + N - \frac{15}{4}$

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

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

$$\frac{\{S_w + S_N\}}{N} = \frac{N+N^2+N-\frac{15}{4}}{N} = 2 + N - \frac{15}{4N} = N - \frac{15}{4}N = N$$

This gives us an amortized runtime complexity of $O(N)$.