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						5					10									
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																			
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:

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

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 our example:

$$S_k = 5\left(\frac{1-2^k}{1-2}\right)$$

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,

$\frac{N}{2} = 5 * 2^{k-1}$

$\Rightarrow \frac{N}{10} = 2^{k-1}$

$\Rightarrow \frac{N}{10} = \frac{2^k}{2}$

$\Rightarrow \frac{N}{5} = 2^k$

$\Rightarrow log_2(\frac{N}{5}) = log_2(2^k)$ [multiplying both sides by log_2]

$\Rightarrow k = log_2(\frac{N}{5})$

$S_N = 5\left(\frac{1-2^{log_2(\frac{N}{5})}}{1-2}\right) = 5\left(\frac{1-(\frac{N}{5})}{-1}\right) = N-5$

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:

$\frac{\{S_w + S_N\}}{N} = \frac{\{N+N-5\}}{N} = \frac{\{2N-5\}}{N} = 2 - \frac{5}{N} \leq 2$ [Because, when N is very large, $\frac{2}{N} \approx 0$]

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						5		7		9		11		13		15		17		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		21		23		25		27		29		31		33		35		37		39
Total Cumulative Cost	117	139	140	164	165	191	192	220	221	251	252	284	285	319	320	356	357	395	396	436

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.

For the sequence in our example:

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

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

For the sequence in our example:

$$S_k = k\left(\frac{5+5+2(k-1)}{2}\right) = k\left(\frac{2k+8}{2}\right) = k\,(k+4) = k^2 + 4k$$

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, $N > 5$)). You can consider either of the cases (i.e., N is odd or N is even). Let's consider N is even (you can consider N is odd too) and we get $a_k = N - 1$. Which gives:

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

$$\Rightarrow N - 1 = 5 + 2k - 2$$

$$\Rightarrow k = \frac{N-4}{2}$$

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

$$\left(\frac{N-4}{2}\right)^2 + 4\left(\frac{N-4}{2}\right) = \left(\frac{N-4}{2}\right)\left[\left(\frac{N-4}{2}\right) + 4\right] = \left(\frac{N-4}{2}\right)\left(\frac{N-4+8}{2}\right) = \frac{(N^2 - 16)}{4} = \left(\frac{N^2}{4} - 4\right)$$

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:

$$\frac{\{S_w + S_N\}}{N} = \frac{\{N + (\frac{N^2}{4} - 4)\}}{N} = 1 + \frac{N}{4} - \frac{4}{N} \leq N$$ [When N is very large, constants can be ignored and $\frac{4}{N} \approx 0$]

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