CS-261: Assignment 2 Written Analysis

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

a)

Append	1 1	2 2	3 3	4 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				3			6						12							
Total Cost	1	2	3	**7**	8	9	**16**	17	18	19	20	21	**34**	35	36	37	38	39	40	41

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					24															
Total Cost	42	43	44	45	**70**	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85

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: 3, 6, 12, ... (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:

$a_k = 3 * 2^{k-1}$

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:

$$S_k = 3 \left(\frac{1-2^k}{1-2}\right)$$

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,

$\frac{N}{2} = 3 * 2^{k-1}$

$\Rightarrow \frac{N}{6} = 2^{k-1}$

$\Rightarrow \frac{N}{6} = \frac{2^k}{2}$

$\Rightarrow \frac{N}{3} = 2^k$

$\Rightarrow log_2(\frac{N}{3}) = log_2(2^k)$ [multiplying both sides by log_2]

$\Rightarrow k = log_2(\frac{N}{3})$ [because $log_2(2)^k = k$]

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

$S_N = 3\left(\frac{1-2^{log_2(\frac{N}{3})}}{1-2}\right) = 3\left(\frac{1-(\frac{N}{3})}{-1}\right) = N-3$

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:

$\frac{\{S_w + S_N\}}{N} = \frac{\{N+N-3\}}{N} = \frac{\{2N-3\}}{N} = 2 - \frac{3}{N} \leq 2$ [Because, when N is very large, $\frac{3}{N} \approx 0$]

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				3		5		7		9		11		13		15		17		19
Total Cost	1	2	3	7	8	14	15	23	24	34	35	47	48	62	63	79	80	98	99	119

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 Cost	120	142	143	167	168	194	195	223	224	254	255	287	288	322	323	359	360	398	399	439

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:

3, 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:

$$a_k = 3 + 2\,(k-1)$$

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:

$$S_k = k\left(\frac{3+3+2(k-1)}{2}\right) = k\left(\frac{2k+4}{2}\right) = k\,(k+2) = k^2 + 2k$$

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). Let's consider N is even (you can consider N is odd too) and we get $a_k = N - 1$. Which gives:

$$N - 1 = 3 + 2(-1)$$

$$\Rightarrow N - 1 = 3 + 2 - 2$$

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

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

$$S_N = \left(\frac{N-2}{2}\right)^2 + 2\left(\frac{N-2}{2}\right) = \left(\frac{N^2}{4} - 1\right)$$

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:

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

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