Julian Garcia
CSE 344: Hw4

I certify that every answer in this assignment is the result of my own work; that Ihave neither copied off the Internet nor from any one else's work; and I have not shared my answers or attempts at answers with anyone else.

1. A data structure D supports a single operation OP. A sequence of n OP operations is performed on D. The actual cost of the i^{th} operation $(1 \leq i \leq n)$ is i if i is an exact power of 2 and 1 otherwise.

   Using the method of aggregate analysis, show that the amortized cost of the above sequence of n operations is $O(n)$.

Let c_i represent the cost:
c_i = i if i is a power of 2, 1 otherwise

Let's create a table of i vs c_i

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c_i	1	2	1	4	1	1	1	8	1	1	1	1	1	1	1	16

We can conclude here that the number of operations costing 1 is < n, and that 2^i instances occur logarithmically.
So we can derive this inequality:

$$\sum_{i=1}^{n} c^i \leq n + \sum_{j=0}^{m} 2^j$$

Where m = (log(n))
Both terms on the right hand side of the inequality are O(n), so the total running time of n insertions is O(n). Therefore, the amortized cost of the sequence of n OP operations is O(n).

2. A data structure D supports a single operation OP. A sequence of n OP operations is performed on D. The actual cost of the i^{th} operation $(1 \leq i \leq n)$ is i if i is an exact power of 2 and 1 otherwise.

   Using the accounting method of analysis (where we pretend that the unit of cost is a dollar), find the minimum x so that by charging a flat fee of \$$x$ for OP you can prove that the amortized cost of a sequence of n operations is $O(n)$.

   Hint: Your strategy should be to earn enough profit during the cheap (\$1) spell so as to entirely pay for the subsequent (\$$2^j$) spike.

For the x, it's easy to see that charging a flat rate of 1 wouldn't add up enough to cover the cost of the \$$2^j$ spike. A rate of two would also definitely fail at the 8th index, so we'll try a rate of three.

Let's revisit the table I made for the first problem and add two rows, one for cost (denoted by $x) and one for balance at index i (denoted by B_i):

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c_i	1	2	1	4	1	1	1	8	1	1	1	1	1	1	1	16
$x	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
B_i	2	3	5	4	6	8	10	5	7	9	11	13	15	17	19	6

So we can see that this value for x works from 1 to 16

Now let j > 0
After 2^{j-1} operations there is one unit of credit. Between operations 2^{j-1} and 2^j there are 2^{j-1-1} operations none of which is an exact power of 2. Each assigns three units as credit resulting to total of $1+3 \cdot (2^{j-1} - 1) = 2^j - 1$ accumulated credit before the 2^j operation. This is just enough to cover its true cost.

Opn	Actual Cost	Assigned Amortized Cost
OP	i if i is a power of 2, 1 otherwise	3

Total amortized cost of sequence = 3n
Hence actual cost is O(n)