Assignment 1

Siwei Tang (siweitang)
sw.tang@mail.utoronto.ca

Changyan Xu (xuchangy)
changyan.xu@mail.utoronto.ca

April 15, 2021

Reference

Not applicable.

Question 3

a) Let k+1th operation be the one that causes bankruptcy.

Since each time after 10 INSERT operations, the data structure needs to resize that cost many credits for coping previous elements into new structure, so in order for bankruptcy, k would be multiple of 10. In such a case, k+1 would cause bankruptcy.

Total credit: since each INSERT operation charges \$5 and costs \$1 for array access. So 4 credits left for every INSERT operation. Since there are k+1 insert operations, so there are total k+1 elements. So total input credits are \$4(k+1).

For the credit consumed during resize(copy takes \$2):
Before 11th INSERT operation, the first resize, we need to copy the previous 10 elements, so costs 2×10
Before k+1th INSERT operation, the $\frac{k}{10}$th resize, we need to copy the previous k elements, so costs $2 \times \frac{k}{10} \times 10$

$$2 \times 10 + 4 \times 10 + ... + \frac{k}{10} \times 2 \times 10 = 20 \sum_{i=1}^{\frac{k}{10}} i = \frac{k^2}{10} + k$$

Let the cost larger than the credit (bankruptcy):

$$\frac{k^2}{10} + k > 4k + 4$$
$$\frac{k^2}{10} - 3k - 4 > 0$$
$$k^2 - 30k - 40 > 0$$
$$k = \frac{30 + \sqrt{900 + 160}}{2} \approx 31.28(k \textit{ is greater than } 0)$$

since k is multiple of 10 and $k^2 - 30k - 40 > 0$
$\Rightarrow$ k = 40
$\Rightarrow$ 40 INSERT operations will take <u>before</u> the data structure runs out of money.

b) For the first b elements, INSERT charges \$3 and costs \$1 for access, so 2 credits left for each element, total \$2b. For the first resize, we need to copy the b elements into the new structure, each cost \$2, so total \$2b which is the same amount as the credits.

For the second b elements, INSERT charges \$5 and costs \$1 for access, so 4 credits left for each element, total \$4b. For the second resize, we need to copy the 2b elements into the new structure, each cost \$2, so total \$4b which is the same amount as the credits.

For the dth b elements, INSERT charges \$(3+2d) and costs \$1 for access, so 2d+2 credits left for each element, total \$(2d+2)b. For the (d+1)th resize, we need to copy the (d+1)b elements into the new structure, each cost \$2, so total \$(2d+2)b which is the same amount as the credits.

So we can find that the total credit after dth resize consists of INSERT from (bd+1)th element to (bd+b)th element.

$\Rightarrow$ CI(credit Invariant) = 2(d + 1)(b - e)
d = number of resize
e = number of empty blocks

Prove by induction, Let P(d): CI(d) = 2(d + 1)(b - e_d)

Base Case:
d = 0, e = b, credit = 0(which means no INSERT)
P(0): CI = 0 = 2(0 + 1)(b - b)
$\Rightarrow$ P(0) holds

Inductive Step:
Assume P(d) holds i.e. CI(d) = 2(d + 1)(b - e_d), want to prove P(d + 1) holds

Case 1: INSERT that would not cause resize
Want to Prove: P(d + 1): CI(d + 1)= 2(d + 1)(b - e_{d+1})
For the new INSERT element, charge \$3+2d and costs \$1 so \$2d+2 credits left.
$\Rightarrow$CI(d + 1) = CI(d) + (2d + 2)
e_{d+1} = e_d - 1, d does not change
By Inductive hypothesis,CI(d) = 2(d + 1)(b - e_d)
$\Rightarrow$CI(d + 1) = CI(d) + (2d + 2) = 2(d + 1)(b - e_d) + 2(d + 2) = 2(d + 1)(b - e_d + 1) = 2(d + 1)(b - e_{d+1})
$\Rightarrow$ P(d + 1) holds

Case 2: INSERT that would cause (d + 1)th resize
Want to Prove: P(d + 1): CI(d + 1)= 2(d* + 1)(b - e_{d+1})
In such a case, $e_d = 0$ since only when structure is full, we need to resize for new INSERT
For the new INSERT element, charge \$3 + 2(d + 1) and costs \$1 so \$2d + 4 credits left.
For the copy cost during (d + 1)th resize, it would cost \$2(d + 1)b. $\Rightarrow$ CI(d + 1) = CI(d) - 2b(d + 1) + 2d + 4
$e_{d+1} = b - 1$ since it is the first INSERT element after resize.
d* = d + 1 since we have one more resize.
By Inductive hypothesis,CI(d) = 2(d + 1)(b - e_d)
$\Rightarrow$CI(d + 1) = CI(d) - 2b(d + 1) + 2d + 4
$\Rightarrow$ CI(d + 1)= 2(d + 1)(b - e_d) - 2b(d + 1) + 2d + 4 = 2(d + 1)b - 2b(d + 1) + 2d + 4 = 2d + 4
$\Rightarrow$ CI(d + 1) = 2(d + 2) $\times$ 1 = 2(d + 2)(b - e_{d+1} = 2(d* + 1)(b - e_{d+1})
$\Rightarrow$ P(d + 1) holds.

c) Let $k \in \mathbb{N}$, denote t_k as the cost for insert kth element.

When $k = ab + 1$ where $b \in \mathbb{N}$, $t_k = 2ab + 1$; otherwise t_k = 1.

For m insert operations, denote the total cost $T(m) = \sum_{k=1}^{m} t_k$

Since every insert operation costs at least \$1, let $t'_k = t_k - 1$

$$
\begin{aligned}
T(m) &= \sum_{k=1}^{m} t_k \\
&= \sum_{k=1}^{m} t'_k + m \\
&= \sum_{a=1}^{\lfloor \frac{m-1}{b} \rfloor} t'_{ab+1} + m \\
&= \sum_{a=1}^{\lfloor \frac{m-1}{b} \rfloor} 2ab + m \\
&= 2b \times \sum_{a=1}^{\lfloor \frac{m-1}{b} \rfloor} a + m \\
&= 2b \times \frac{(1 + \lfloor \frac{m-1}{b} \rfloor) \times \lfloor \frac{m-1}{b} \rfloor}{2} + m \\
&= b \times \lfloor \frac{m-1}{b} \rfloor^2 + b \times \lfloor \frac{m-1}{b} \rfloor + m
\end{aligned}
$$

For a single insert operation, it is

$$
\begin{aligned}
T &= \frac{T(m)}{m} = \frac{b \times \lfloor \frac{m-1}{b} \rfloor^2 + b \times \lfloor \frac{m-1}{b} \rfloor + m}{m} \\
\Rightarrow T &\in \Theta(m)
\end{aligned}
$$