Illinois Institute of Technology
Department of Computer Science

Solutions to Homework Assignment 5

CS 430 Introduction to Algorithms
Spring Semester, 2018

Solution:

1. For a TABLE-DELETE operation that does not incur contraction, the amortized cost is:

$$\begin{aligned}\widehat{c_i} = c_i + \Phi(D_i) - \Phi(D_{i-1}) &\leq 1 + |2 \cdot num_i - size_i| - |2 \cdot num_{i-1} - size_{i-1}| \\ &= \begin{cases} 1 + 2(num_i - num_{i-1}) = -1 & 2num_i \geq size_i \\ 1 - 2(num_i - num_{i-1}) = 3 & 2num_{i-1} \leq size_i \\ 1 + 2size_{i-1} - 2(num_i + num_{i-1}) & 2num_i < size_i < 2num_{i-1} \end{cases}\end{aligned}$$

   The term in the third case is equal to

$$1 + 2size_{i-1} - 2(num_{i-1} - 1 + num_{i-1}) = 3 + 2(size_{i-1} - 2num_{i-1}) < 3$$

   because $size_{i-1} = size_i < 2num_{i-1}$ (case 3 condition). Therefore, the upper bound in this case is 3, which is $O(1)$.

   For a TABLE-DELETE operation that incurs contraction, the amortized cost is:

$$\begin{aligned}\widehat{c_i} = c_i + \Phi(D_i) - \Phi(D_{i-1}) &\leq num_i + 1 + |2 \cdot num_i - size_i| - |2 \cdot num_{i-1} - size_{i-1}| \\ &= num_i + 1 + |2 \cdot num_i - \frac{2}{3}size_{i-1}| - |2 \cdot num_{i-1} - size_{i-1}| \\ &= num_i + 1 + 0 - |2 \cdot num_{i-1} - size_{i-1}| \qquad (num_i = \frac{1}{3}size_{i-1}) \\ &= num_i + 1 + 2 \cdot num_{i-1} - size_{i-1} \qquad (2num_{i-1} < size_{i-1}) \\ &= 3 \in O(1)\end{aligned}$$

2. ** Note that there may be different deletion algorithms, which lead to different cost and different potential functions. The solution here is just one of them.

   (a) When inserting an element at the head, push the element to the *Head*. When inserting an element at the tail, push the element to the *Tail*.

   (b) Without the loss of generality, I assume it is the *Tail* stack which is empty. For the case where *Head* is empty, it is trivial to get the similar algorithm based on the following one.

      i. Iteratively POP each element from *Head* and PUSH it into *Temp* until the last element.

      ii. POP the last element in *Head* and discard it.

      iii. In the remaining $n-1$ elements in *Temp*, do the following to the first $(n-1)/2$ elements:

         A. Iteratively POP and PUSH each of them to *Head*.

         B. Iteratively POP and PUSH every element from *Head* to *Tail*.

      iv. For the remaining $(n-1)/2$ elements in *Temp*, do the following: iteratively POP and PUSH each of them to *Head*.

(c) **Insertion**

In any case (either best, average or worst), the complexity of insertion is $O(1)$.

Deletion

In fact, the worst case is when we need to delete an element from the tail but *Tail* is empty or we need to delete an element from the head but *Head* is empty. Then, we can simply look at the above algorithm to find out the worst-case cost.

Suppose the costs of POP and PUSH are both 1, then the step 1's cost is $2(n-1)$. The cost of step 2 is 1. The cost of 3.(a) is $(n-1)/2 \times 2 = n-1$. The cost of 3.(b) is $(n-1)/2 \times 2 = n-1$. The cost of 4 is $(n-1)/2 \times 2 = n-1$. Sum of the cost above is $5(n-1)+1 = 5n-4$.

(d) **Wost-case deletion**

Similarly, I assume it is *Tail* stack who is empty. Suppose the potential function is $\Phi(D_i) = k\Big||Head_i| - |Tail_i|\Big|$, where $Xxxx_i$ refers to the corresponding stack after the i-th operation. Then, the amortized cost is:

$$
\begin{aligned}
\widehat{c_i} = c_i + \Phi(D_i) - \Phi(D_{i-1}) =& 5n - 4 + k\Big||Head_i| - |Tail_i|\Big| - k\Big||Head_{i-1}| - |Tail_{i-1}|\Big| \\
&= 5n - 4 + k\Big||(n-1)/2| - |(n-1)/2|\Big| - k\Big|n - 0\Big| \\
&= 5n - 4 - kn
\end{aligned}
$$

which should be $O(1)$. Therefore, $k = 5$, and the potential function is $\Phi(D_i) = 5\Big||Head_i| - |Tail_i|\Big|$.

Normal deletion and insertions

In any case, it is easy to derive that the amortized time of deletion in the normal case is between $[1-k, 1+k]$, which is always constant. The amortized time of both insertions is same. Therefore, k can be any number for normal deletion or insertions.

Conclusion

In conclusion, k has to be 5 to have constant amortized time for all four operations.