Problem Set 6

Due date: Electronic submission of the pdf file of this homework is due on **3/8/2024 before 11:59pm** on canvas.

Name: Jawahar Sai Nathani

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. This work is my own. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework. Looking up solutions is prohibited.

Signature: Jawahar Sai Nathani

Make sure that you describe all solutions in your own words, and that the work presented is your own! Typesetting in LaTeX is required. Read the chapter on amortized analysis in our textbook.

Problem 1 (20 points)**.** In the lecture, we discussed a stack with PUSH, POP, and MULTIPOP operations. If the set of stack operations included a MULTI-PUSH operation, which pushes k items onto the stack, would the $O(1)$ bound on the amortized cost of stack operations continue to hold?

Solution. No, if the set of stack operations includes a Multi-Push operation, then the $O(1)$ bound on the amortized cost of stack operations would not continue to hold.

- When k items are pushed onto the stack with a single MULTI-PUSH operation, the cost of this operation is proportional to k, not to 1.
- Therefore, if k is large or varies significantly, the amortized cost of stack operations may no longer be constant. The amortized cost per operation in this scenario might be $O(k)$.

Problem 2 (20 points)**.** In the lecture, we discussed a k-bit counter with an INCREMENT operation. Show that if a DECREMENT operation were included in the k-bit counter example, n operations could cost as much as $\Theta(nk)$ time.

Solution. Let's analyze the worst-case scenario where we perform n operations on a k-bit counter, consisting entirely of alternating INCREMENT and DECREMENT operations.

- For each INCREMENT operation, we start with the least significant bit (LSB) and flip the first 0 to 1. This requires scanning through k bits if the first 0 is at the start of the counter.
- Similarly, for each DECREMENT operation, we start with the least significant bit (LSB) and flip the first 1 to 0. This also requires scanning k bits if the first 1 is at the start of the counter.
- Therefore, In the worst-case scenario, both INCREMENT and DECREMENT operations may need to scan through all k bits of the counter. This leads to a time complexity of $O(k)$ per operation.
- Therefore, for n operations, the total time complexity could be as much as $\Theta(nk)$.

Hence, if a DECREMENT operation were included in the k-bit counter-example, in the worst-case scenario n operations could cost as much as $\Theta(nk)$.

Problem 3 (20 points)**.** Suppose we perform a sequence of n operations on a data structure in which the i-th operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

Solution. To calculate the amortized cost per operation using aggregate analysis, we first calculate the total cost for n operations and divide it by n.

Total Amortized cost

- If i is not an exact power of 2, the cost is 1. If i is an exact power of 2, the cost is i which can also be written as 2^k.
- In n number of operations, there are $\lceil log_2 n \rceil$ exact powers of 2.
- Let's consider the cost of an operation i is $c(i)$. Sum of $c(i)$ would be

$$\begin{aligned}\sum_{i=1}^{n} c(i) &= \sum_{k=1}^{\lceil \lg n \rceil} 2^k + \sum_{i \leq n \text{ is not a power of 2}} 1 \\ &\leq \sum_{k=1}^{\lceil \lg n \rceil} 2^k + n \\ &= 2^{1+\lceil \lg n \rceil} - 1 + n \\ &\leq 2n - 1 + n \\ &\leq 3n \in O(n).\end{aligned}$$

Therefore, amortized cost per operation is $\frac{O(n)}{n}$, that is $O(1)$.

Problem 4 (20 points)**.** Redo Problem 3 using an accounting method of analysis.

Solution. Let c_i = cost of i th operation.

$$c_i = \begin{cases} i & \text{if } i \text{ is an exact power of 2,} \\ 1 & \text{otherwise.} \end{cases}$$

Charge 3 (amortized cost $\hat{c}_i$) for each operation.

- If i is not an exact power of 2 , pay \$1, and store \$2 as credit.
- If i is an exact power of 2 , pay \$$i$, using stored credit.
- Let now $j > 0$. After 2^{j-1} th operation 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 two units as credit resulting in a total of $1 + 2 \cdot \left(2^{j-1} - 1\right) = 2^j - 1$ accumulated credit before 2^j th operation. This, together with one unit on its own, is just enough to cover its true cost.
- Therefore the amount of credit stays non-negative all the time

Since the credit never goes negative and the amortized cost of each operation is 3 - $O(1)$, the amortized cost of n operations is $O(n)$.

Problem 5 (20 points)**.** Redo Problem 3 using a potential method of analysis.

Solution. Let the potential function be

$$\Phi\left(D_0\right)=0 \quad \text{and} \quad \Phi\left(D_i\right)=2i-2^{1+\lfloor \ln i \rfloor} \text{ for } i>0$$

- For operation 1,

$$\begin{aligned}
\hat{c}_1 &= c_1+\Phi\left(D_1\right)-\Phi\left(D_0\right)\\
&= 1+2-2^{1+\lfloor \ln 1 \rfloor}-0\\
&= 1
\end{aligned}$$

- For operation $i(i>1)$, if i is not a power of 2 , then

$$\begin{aligned}
\hat{c}_i &= c_i+\Phi\left(D_i\right)-\Phi\left(D_{i-1}\right)\\
&= 1+2i-2^{1+\lfloor \ln i \rfloor}-\left(2(i-1)-2^{1+\lfloor \ln (i-1) \rfloor}\right)\\
&= 1+2i-2i-2^{1+\lfloor \ln i \rfloor}+2^{1+\lfloor \ln (i-1) \rfloor}+2\\
&= 3
\end{aligned}$$

- If $i=2^j$ for some $j \in \mathbb{N}$, then

$$\begin{aligned}
\hat{c}_i &= c_i+\Phi\left(D_i\right)-\Phi\left(D_{i-1}\right)\\
&= i+2i-2^{1+\lfloor \ln 2^j \rfloor}-\left(2(i-1)-2^{1+\lfloor \ln (2^j-1) \rfloor}\right)\\
&= i+2i-2^{1+j}-\left(2(i-1)-2^{1+j-1}\right)\\
&= i+2i-2i-2i+2+i\\
&= 2
\end{aligned}$$

Therefore, the amortized cost of each operation is $O(1)$.

Discussions on canvas are always encouraged, especially to clarify concepts that were introduced in the lecture. However, discussions of homework problems on canvas should not contain spoilers. It is okay to ask for clarifications concerning homework questions if needed. Make sure that you write the solutions in your own words.

Checklist:
✓ Did you add your name?
✓ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
✓ Did you sign that you followed the Aggie honor code?
✓ Did you solve all problems?
✓ Did you submit the pdf file of your homework?