CIS 675 (Fall 2018) Disclosure Sheet

Name: Wentan Bai
HW # 6

(Yes) **No** Did you consult with anyone on parts of this assignment, including other students, TAs, or the instructor?

Yes **(No)** Did you consult an outside source (such as an Internet forum or a book other than the course textbook) on parts of this assignment?

If you answered **Yes** to one or more questions, please give the details here:

I also consult all questions and extra question with another student, Wentian Bai. For all questions, we discussed our ideas and I finish my algorithms independently.

By submitting this sheet through my Blackboard account, I assert that the information on this sheet is true.

Question 1:

If we ignore the new costs, the average cost per increment is $\$2$ per opration, as the same as the original in-class problem.

The cost of initialization is $\$2^i$ to initialize a bit at the i_{th} position. For n operations, the total running time is $2^0 + 2^1 + 2^2 + ... + 2^{\log_2 n} = 2^{\log_2 n+1} - 1 \approx 2 \cdot 2^{\log_2 n} = 2n$

Thus the final total running time is $2n + 2n = 4n$, and pre increments costs $\frac{4n}{n} = \$4/operation$

Question 2:

Each Push operation pays $\$2$, where $\$1$ for push operation itself and $\$1$ for bank.
Each pop operations also pays $\$2$, where $\$1$ for pop operation itself and $\$1$ for bank.
Then at every K operations, we use dollar in bank to pay for copy.
Thus, the average cost per operation is $\$2$

Question 3:

Each Push operation pays $\$3$, where $\$1$ for push operation itself and $\$2$ for bank.

Each pop operation costs $\$1$ from bank.

When stack reaches k elements, there is at least $\$2k$ in bank, so we use dollars in bank to pay for copy and delete all elements from the stack.

Thus, the upper bound on the average cost per operation is $\$3$.

Question 4 a:

No. If there are n elements and the array is full, we add one element and then remove last one, then add one element and then remove last one, and keep repeating this step. It means that we will need to expand array, shrink array, expand array, shrink array, and repeat this two steps. In this sequence, there are not cheap operations, so the amortized analysis does not make sense here.

Question 4 b:

The insertion operation still needs $\$3$, $\$1$ for insertion operation itself and $\$2$ for bank.

The remove operation needs $\$2$, $\$1$ for remove operation itself and $\$1$ for bank.

When we expand or shrink array, we use dollars in bank to copy elements.

Thus, the average cost per operation is larger than $\$2$ and less than or equal to $\$3$. The upper bound on the average cost per operation is $\$3$.

Extra:

If k is a power of 2, we pay $\$2$. Otherwise, we pay $\$3$.

To using the banking method, we need to ensure the bank will neven go bankrupt. When k is not a power of 2, payment is enough to cover the cost of operation. When k equals to 1 or 2, the payment is also enough to cover the cost of operation.

Assume k is a power of 2, denoted as 2^i. The numbers between 2^i and 2^{i-1} is $2^i - 2^{i-1} = 2^{i-1}$. We need to use 2^{i-1} times to accumulate enough money to pay $\$2^i$. Thus, if k is not a power of 2, we need to pay $\$2$ for bank and the $\$1$ for cover the cost of operation.

Therefore, the upper bound on the average cost per operation is $\$3$.