CSC225 - Assignment 2
Nelson Dai - V00815253
2016-06-06

1. In order to implement the stack ADT, Let's name these two queues q1 and q2. We add new items in the back of the queue but set the front item as the top of the list. q2 will just be an temporary storage when we add new items.

Pop: Since we set the front item as the top of the list, when we dequeue it would remove the front item just like stack.

Push: Assume the item that we would like to push is E. First, we need to dequeue all the items from q_1 and enqueue them all into q_2 one by one. Second, we enqueue the new item E into q_1. Finally, we dequeue all items from q_2 and store them back to q_1 so that the new item is in the front (top) of the list.

run time: Pop is O(1), Push is O(n)

2. Selection sort:

{5, 7, 0, 3, 4, 2, 6, 1}
→ {0, 7, 5, 3, 4, 2, 6, 1}
→ {0, 1, 5, 3, 4, 2, 6, 7}
→ {0, 1, 2, 3, 4, 5, 6, 7}

Bubble Sort:

{5, 7, 0, 3, 4, 2, 6, 1}
→ {5, 0, 3, 4, 2, 6, 1, 7}
→ {0, 3, 4, 2, 5, 1, 6, 7}
→ {0, 3, 2, 4, 1, 5, 6, 7}
→ {0, 2, 3, 1, 4, 5, 6, 7}
→ {0, 2, 1, 3, 4, 5, 6, 7}
→ {0, 1, 2, 3, 4, 5, 6, 7}

Insertion Sort:

{5, 7, 0, 3, 4, 2, 6, 1}
→ {0, 5, 7, 3, 4, 2, 6, 1}
→ {0, 3, 5, 7, 4, 2, 6, 1}
→ {0, 3, 4, 5, 7, 2, 6, 1}
→ {0, 2, 3, 4, 5, 7, 6, 1}
→ {0, 2, 3, 4, 5, 6, 7, 1}
→ {0, 1, 2, 3, 4, 5, 6, 7}

3. For i from 0 to n-1
Add i to sum
For i from 0 to n-1
Add Array[i] to sum_of_Array
missingNumber = sum - sum_of_Array.

-1 Didn't explain the time complexity and space complexity

4. In both cases we assume that the array's length is increased k times.

$f(n) = N + c$: let $(n-1) = k$, $1+2+3+\cdots+(n-1) = \frac{(n-1)n}{2} = \frac{n^2-n}{2} = O(n^2)$

$f(n) = 2N$: we know that $N \leq 2^k \rightarrow \log_2 N \leq k$

$1+2+4+8+\cdots+2N \rightarrow 2^{\log_2 N + 1} - 1 \rightarrow 2N - 1 = O(N)$

since $O(N)$ grows much slower than $O(N^2)$ so double the array size each time is better than adding c.

5. $T(n) = 2T(n/2) + \log n$

$n^{\log_b a} = n^{\log_2 2} = n \rightarrow f(n) = \log n = O(n^{1-\epsilon})$ (when $\epsilon = \frac{1}{2}$) $\therefore$ Case 1 $T(n) = \Theta(n)$

$T(n) = 8T(n/2) + n^2$

$n^{\log_b a} = n^{\log_2 8} = n^3 \rightarrow f(n) = n^2 = O(n^{3-\epsilon})$ (when $\epsilon = \frac{1}{2}$). $\therefore$ Case 1 $T(n) = \Theta(n^3)$

$T(n) = 16T(n/2) + (n \log n)^4$

$n^{\log_b a} = n^{\log_2 16} = n^4 \rightarrow f(n) = (n \log n)^4 = n^4 (\log^4 n)$ $\therefore$ case 2 $T(n) = \Theta(n^4 (\log n)^5)$

$T(n) = 7T(n/3) + n$

$n^{\log_b a} = n^{\log_3 7} \approx n^{1.77} \rightarrow f(n) = n = O(n^{1.77})$ $\therefore$ case 1 $T(n) = \Theta(n^{1.77})$

$T(n) = 9T(n/3) + n^3 \log n$

$n^{\log_b a} = n^{\log_3 9} = n^2 \rightarrow f(n) = n^3 \log n = \Omega(n^{2+\epsilon})$ (when $\epsilon = 1$)

$\because af(n/b) = 9(n/3)^3 \log(n/3) = n^3/3 \log(n/3) \leq \delta f(n)$ for $\delta = \frac{1}{3}$ and $n \geq 1$

$\therefore$ case 3 $T(n) = \Theta(n^3 \log n)$