HW #5

Chris Maccom

17.4-3

Looking at p. 476 in CLRS, suppose the amortized cost of TABLE_DELETE depends on if the table resizes.

insertion & deletion have similar time complexities, hence we can aggregate the insert function to set the followings:

if no resize is triggered: $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

$= 1 + |2 \cdot num_i - size_i| - |2(num_i - 1) - size_i|$

$= 3$

With no resize, the amortized cost is bound by a constant of 3.

if the load factor is $< \frac{1}{3}$, a resize is triggered such that: $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$

$\leq num_i + 1 + \frac{2}{3} size_{i-1} - 2num_i - size_{i-1} - 2num_i - 2$

$= num_i + 1 + 2num_i + 2 - 2num_i - 3num_i - 3 + 2num_i$

$= 2$

note
c_i is i if $i-1$ is an exact power of 2
see page 465

with the load factor drops below $\frac{1}{3}$ and its size compresses to $\frac{2}{3}$, the function is bound by a constant of 2

2) (a) A deque is essentially a set of two stacks:
- a head stack which allows push() and pop() from the head of the head stack
- a tail stack which allows push() and pop() from the head of the tail stack.

This takes O(1) time since the operations only index one var.

(b) If a stack is empty pop half the elements from the non empty stack and push them to the temp stack.

The elements in the Temp stack are in reverse order.

Then pop the remaining elements and push them into the other (non-Temp) stack. This reverses the order and empties the original stack.

Then, pop all elements from temp and push them onto the non-empty stack, which will transfer the remaining elements. Delete the necessary element by not pushing it in this, or the above step.

(c) Insert ~~/delete~~ from the front of the head or tail stack is O(1). if one of the two stacks is empty, and the last element needs to be deleted ~~or inserted at~~, the time complexity is O(n).

(d) Based on the above case, insert operations will take O(1) time delete operations with non-identical head & tail stacks take $O(||Head| - |Tail||)$ time in the worst case

$$\hat{C} = \frac{1}{n}\sum_{i=1}^{n} cost(o_i) = \begin{cases} \Theta(1) & \text{for insert} \\ \Theta(||H| - |T||) & \text{for delete} \end{cases}$$