1. Amortized Analysis
   - Your task is to prove that a dynamic table's insertion and deletion operations take constant time in amortized analysis, considering that the table may expand or contract after the operation.
   - While the class slide covers some cases, some non-critical ones are left out. Therefore, your job is to complete all the cases and submit a handwritten proof of this problem.

a.

Table Insertion:

In table insertion, the ith operation causes an expansion only when $i-1$ is an exact power of 2. The amortized cost of an operation is in fact $O(1)$, we can show in aggregate analysis.

The cost of the ith operation is

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

The total cost of n table insert operations is therefore

$$\sum_{i=1}^{n} c_i \leq n + \sum_{j=0}^{\lfloor \lg n \rfloor} 2^j$$
$$< n + 2n$$
$$= 3n \;=\; O(n)$$

because at most n operations cost 1 and the costs of the remaining operations form a geometric series. Since the total cost of n table insert operations is bounded by $3n$, the amortized cost of a single operation is at most 3.
So the time complexity is $O(1)$.

Table Deletion:

By using potential method, the amortized cost for deletion is

$$\hat{c_i} = c_i + \phi_i - \phi_{i-1}$$
$$= 1 + 2N_i - S_i - (2N_{i-1} - S_{i-1})$$
$$= 1 + 2(N_{i-1} - 1) - S_i - 2N_{i-1} + S_i \quad \text{since } S_i = S_{i-1},\ N_i = N_{i-1} - 1$$
$$= -1$$

where S_i is the table size of the ith operation, and N_i is the number of elements in the ith operation.

Thus, the amortized cost of delete is $-1 = O(1)$

b.

The potential function is define as.

$$\phi(T) = \begin{cases} 2 \cdot T.num - T.size & \text{if } \alpha \geq \frac{1}{2} \\ T.size/2 - T.sum & \text{if } \alpha < \frac{1}{2}. \end{cases}$$

* ith operation is insertion.

case 1: $\alpha_{i-1} \geq \frac{1}{2} \Rightarrow \alpha_i \geq \frac{1}{2}$.

$$\hat{c_i} = 1 + 2num_i - size_i - 2(num_i - 1) + size_{i-1}$$
$$= 1 + 2num_i - size_i - 2num_i + 2 + size_i$$
$$= 3.$$

case 2: $\alpha_{i-1} < \frac{1}{2} \rightarrow \alpha_i < \frac{1}{2}$

$$\hat{c}_i = 1 + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1})$$
$$= 1 + size_i/2 - num_i - size_i/2 - num_i - 1$$
$$= 0.$$

case 3: $\alpha_{i-1} < \frac{1}{2} \rightarrow \alpha_i \geq \frac{1}{2}$

$$\hat{c}_i = 1 + 2num_i - size_i - (size_{i-1}/2 - num_{i-1})$$
$$= 1 + 2num_{i-1} + 2 - size_{i-1} - size_{i-1}/2 + num_{i-1}$$
$$= 3 + 3num_{i-1} - 3size_{i-1}/2$$
$$< 3 + 3size_{i-1}/2 - 3size_{i-1}/2$$
$$= 3$$

* ith operation is delete

case 4: $\alpha_{i-1} \geq \frac{1}{2} \rightarrow \alpha_i \geq \frac{1}{2}$

$$\hat{c}_i = 1 + 2num_i - size_i - (2num_{i-1} - size_{i-1})$$
$$= 1 + 2num_i - size_i - 2num_i - 2 + size_i$$
$$= -1$$

case 5: $\alpha_{i-1} \geq \frac{1}{2} \rightarrow \alpha_i < \frac{1}{2}$

$$\hat{c}_i = 1 + (size_i/2 - num_i) - (2num_{i-1} - size_{i-1})$$
$$= 1 + size_i/2 - num_{i-1} + 1 - 2num_{i-1} + size_i$$
$$= 2 + 3size_i/2 - 3num_{i-1}$$
$$= 2 + 3size_i/2 - 3num_i - 3 = -1$$

if $\alpha_{i-1} = \frac{1}{4}$ trigger a contraction

$size_i = size_{i-1}/2$, $num_i = size_{i-1}/2$ and $c_i = num_i + 1$

$$\hat{c}_i = 1 + num_i + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1})$$
$$= 1 + num_i + size_i/2 - num_i - size_i + size_i/2.$$
$$= 1$$

case 6: $\alpha_{i-1} < \frac{1}{2} \rightarrow \frac{1}{4} \leq \alpha_i < \frac{1}{2}$

$$\hat{c}_i = 1 + size_i/2 - num_i - size_{i-1}/2 + num_{i-1}$$
$$= 1 + num_{i-1} + size_i/2 - num_{i-1} + 1 - size_i/2$$
$$= 2.$$

2. Fibonacci Heap

- You are required to show that the degree of any node in a Fibonacci Heap is bounded above by O(log n).

If x is any node of a Fibonacci heap, x.degree $= m$, and x has children $y_1, y_2, \ldots, y_m$, then y_1.degree ≥ 0 and y_i.degree $\geq i-k$. Thus, if S_m denotes the fewest nodes possible in a node of degree m, then we have $S_0 = 1, S_1 = 2, \ldots, S_{k-1} = k$ and in general, $S_m = k + \sum_{i=0}^{m-k} S_i$. Thus, the difference between S_m and S_{m-1} is S_{m-k}.

Let $\{f_m\}$ be the sequence such that $f_m = m+1$ for $0 \leq m < k$ and $f_m = f_{m-1} + f_{m-k}$ for $m \geq k$.

If $F(x)$ is the generating function for f_m then we have $F(x) = \frac{1-x^k}{(1-x)(1-x-x^k)}$. Let α be a root of $x^k = x^{k-1} + 1$. We'll show by induction that $f_{m+k} \geq \alpha^m$. For the base case.

$$f_k = k+1 \geq 1 = \alpha^0$$

$$f_{k+1} = k+3 \geq \alpha^1$$

$$\vdots$$

$$f_{k+k} = k + \frac{(k+1)(k+2)}{2} = k+k+1 + \frac{k(k+1)}{2} \geq 2k+1+\alpha^{k-1} \geq \alpha^k.$$

In general, we have

$$f_{m+k} = f_{m+k-1} + f_m \geq \alpha^{m-1} + \alpha^{m-k} = \alpha^{m-k}(\alpha^{k-1}+1) = \alpha^m$$

Next we show that $f_{m+k} = k + \sum_{i=0}^{m} f_i$. The base case is clear, since $f_k = f_0 + k = k+1$. For the induction step, we have

$$f_{m+k} = f_{m-1-k} + f_m = k\sum_{i=0}^{m-1} f_i + f_m = k + \sum_{i=0}^{m} f_i.$$

Observe that $S_i \geq f_{i+k}$ for $0 \leq i < k$. Again, by induction, for $m \geq k$ we have

$$S_m = k + \sum_{i=0}^{m-k} S_i \geq k + \sum_{i=0}^{m-k} f_{i+k} \geq k + \sum_{i=0}^{m} f_i = f_{m+k}.$$

So in general, $S_m \geq f_{m+k}$. Putting it all together, we have

$$\begin{aligned} \text{size}(x) &\geq S_m \\ &\geq k + \sum_{i=k}^{m} S_{i-k} \\ &\geq k + \sum_{i=k}^{m} f_i \\ &\geq f_{m+k} \\ &\geq \alpha^m \end{aligned}$$

Taking logs on both sides, we have

$$\log_\alpha n \geq m$$

The degree of node is bounded above by $O(\lg n)$