Dynamic Tables

- Dynamically expands and contracts as items are inserted or deleted

- assume new elements are added at the end.

Table Expansion

- when an item is inserted into a full table
  - allocate a larger table
  - copy all items from the old table into the new one
- How large should the new table be?

  Table Doubling

- Allocate a new table with twice the number of slots
- The load factor $\frac{n}{m}$ is always at least $\frac{1}{2}$

```
TableInsertion(T, x)
    if T.num = T.size
        allocate newTable of size T.size ×2
        insert all items into newTable
        T.size = T.size ×2
        free T.table
        T.table = newTable
    insert x into T.table
    T.num = T.num +1
```

$n = T.num$

$m = T.size$

- Cost of the i^{th} operation
  - If the current table has room

    $$c_i = 1$$

  - If the table is full

    $$c_i = i$$

- If we run n operations (Table Insertion), the worst case cost of a single operation is $O(n)$, which leads to a naive $O(n^2)$ bound
- We can get a tighter bound by recognizing that the table rarely expands
  - when $i-1$ is a power of 2

Aggregate Method

- The cost of the i^{th} operation

$$c_i = \begin{cases} 1 \\ i & \text{if } \cancel{i}\ i-1 \text{ is a power of } 2 \end{cases}$$

- The total cost of n Table Insertions

$$\sum_{i=1}^{n} c_i \leq n + \sum_{j=0}^{\lfloor \lg n \rfloor} 2^j$$

$$\sum_{j=0}^{\lg n} 2^j = 2^{\lg n + 1} - 1$$

$$< 2^{\lg n} \cdot 2$$

$$< n + 2n$$

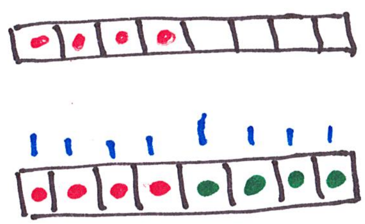
$$= 3n$$

- The amortized cost of a single operation is at most 3 $\Rightarrow$ O(1) amortized cost

Accounting Method

- Each item pays for 3 insertions
  1. Inserting itself into the table
  2. Moving itself into a new table
  3. Moving another item that has already been moved

Example



- when the table is full it contains m items each with $1 credit

- We charge $3 for each insertion
  - $1 for insertion
  - $1 as credit on the new item
  - $1 as credit on an existing item.

Potential Method

- we define our potential function to be 0 immediately after a table expansion and builds to table size by the time the table is full.

$$\Phi(T) = 2\,T.num - T.size$$

- let n_i be the number of items after the i^{th} operation and s_i be the size of the table after the i^{th} operation.
- let Φ_i be the potential after the i^{th} operation.

- If the ith operation does not trigger an expansion $s_i = s_{i-1}$

$$a_i = c_i + \Phi_i - \Phi_{i-1}$$

$$= 1 + (2 \cdot n_i - s_i) - (2n_{i-1} - s_{i-1})$$

$$= 1 + (2n_i - s_i) - (2(n_i - 1) - s_i)$$

$$= 3$$

- If the i^{th} operation does trigger expansion

$$s_i = 2 s_{i-1}$$

$$s_{i-1} = n_{i-1} = n_i - 1$$

$$s_i = 2(n_i - 1)$$

- amortized cost

$$a_i = c_i + \Phi_i - \Phi_{i-1}$$

$$= n_i + (2n_i - s_i) - (2n_{i-1} - s_{i-1})$$

$$= n_i + (2n_i - (2n_i - 2)) - ((2n_i - 2) - (n_i - 1))$$

$$= n_i + 2 - (n_i - 1) = 3$$

- Since our potential function is valid $(\forall i \; \Phi_i \geq \Phi_0)$ the total amortized cost bounds the total actual cost.

Table Contraction

- when the load factor becomes too small we shrink the table.
  - halve the table size when the load factor is $\frac{1}{4}$
  - when the load factor is $\frac{1}{2}$ the potential is 0
- we need the potential to grow to T.num by the time the load factor is $\frac{1}{4}$ or 1

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

- when $\alpha = 1$ $\Phi(T) = T.num$ - when $\alpha = \frac{1}{4}$ $\Phi(T) = T.num$

- Consider a sequence of n TableInsertions and TableDeletions

- If the i^{th} operation is an Insert and $\alpha < \frac{1}{2}$ this operation does not change the table size

$$
\begin{aligned}
a_i &= c_i + \Phi_i - \Phi_{i-1} \\
&= 1 + \left(\frac{s_i}{2} - n_i\right) - \left(\frac{s_{i-1}}{2} - n_{i-1}\right) \\
&= 1 + \left(\frac{s_i}{2} - n_i\right) - \left(\frac{s_i}{2} - (n_i - 1)\right) \\
&= 0
\end{aligned}
$$

- if the i^{th} operation is a Deletion and it does cause a contraction.

$$n_i = n_{i-1} - 1 \qquad \frac{s_i}{2} = \frac{s_{i-1}}{4} = n_{i-1} = n_i + 1$$

$$\begin{aligned} a_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= (n_i + 1) + \left(\frac{s_i}{2} - n_i\right) - \left(\frac{s_{i-1}}{2} - n_{i-1}\right) \\ &= (n_i + 1) + (n_i + 1 - n_i) - (2(n_i + 1) - (n_i + 1)) \\ &= 1 \end{aligned}$$