

Chapter 17

Inserting n elements using

① Aggregate method:

The table doubles its size when it is full. for example if size of ~~an~~ unassigned array is 5, after inserting it double its size to 10.

In general after k doublings the size is 2^k .

Pseudo code:

Initialize table with capacity = 1

for $i = 1$ to n

if table is full:

new table = create new table.

with size $2 * \text{current size}$

Copy elements from old table to new table.

table = new table.

insert element 1 into table.

net, $1 \leq \log(n+1) - 1$

Total cost = $O(n)k$

= $O(n \log n)$

Cost per insertion = $O(\log n)$

Runtime per insertion = $O(\log n)$

Total time is $O(n) * \log(n+1)$.

②

Accounting method.

Charge $2m$ units for each insertion when the table doubles in size from n to $2m$. Credit n units.

$$\text{Total credits} = m + 2m + 4m + \dots$$
$$n/2 * m = O(n)$$

Pseudo Code:

Initialize table with Capacity = 1
for $i=1$ to n

if table is full:

new-table = create new table
with size * current size
copy elements from old table to new
table.
table = new table.

insert element i into table
initialize charges = 0
initialize credits = 0

for $i=1$ to n

charges $t=2$

if table double d in size
from m to $2m$.

Credits $t = m$

Total changes $= 2n = O(n)$

Total credits $= m + 2m + \dots$
 $= n/2 * m = O(n)$

Cost per iteration $= \text{total}/n$
 $= O(n)/n$
 $= O(1)$

Runtime per Insertion $= O(1)$
Total time $= O(n)$.