Dynamic tables

C	Python
array must declare size of array when its defined	initiate an array – keep arbitrarily adding elts

Use analysis we defined yesterday to see that dynamically growing a table is not actually "cheating"

T a table

- .size = # of positions
- .table = pointer to start of table
- .num = # of slots being used

idea - starting with empty table, we fill it in + each time we use up all availible slots, we <u>expand</u> + double the amt. of space availible

Table-insert (T, x)

if T.size = 0
allocate T.table w/ one slot, insert x
T.size = 1

I expansion

if T.num = T.size
allocate new-table of size 2·T.size
insert all elts of T into new-table
free T.table
T.table = new-table
T.size = 2·T.size

insertion

insert x into T.table
T.num ++

Sequence of n table insertions

what's complexity

w/ expansion, we're doing $O(n)$ work

$\rightarrow n \cdot O(n) = O(n^2)$ complexity.

<u>Q</u> how many times do we do an expansion?

<u>ans</u> $\lfloor \log n \rfloor$ because each time we're doubling size of table

Q: bound on size of T in terms of n?

2^n? - no

$T.\text{size} \le 2n$

Obs we're always using $\geq \frac{1}{2}$ T.size slots because we only expand when the table is full + we double the size so at each step, we're using at least $\frac{1}{2}$ available slots.

since we only expand $\log n$ times bound of $O(n \log n)$

Aggregate analysis

c_i = cost of i^{th} insertion

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

$$\sum c_i \le n + \underbrace{\sum_{\substack{i \le n \\ i = 2^j - 1}} i}$$

we have a term here for $j \le \log_2 n$

$$\sum_{i=1}^{n} c_i \le n + \underbrace{\sum_{j=0}^{\log n} 2^{j+1}}_{\le 2n}$$

$$\le 3n$$

$\therefore$ amortized cost of an insertion $= 3 \cdot O(1)$

Accounting method

charge	
3	write in elementary insertion
0	write in the expansion

Bank

each insertion w/out expansion

banks 2

bank has

$$2\left(\underbrace{T.num - \frac{T.size}{2}}_{\text{exactly \# of insertions since the last expansion}}\right)$$

$$= 2\,T.num - T.size$$

show this holds by induction & is ≥ 0

potential method

$$\Phi(D_i) = 2 \cdot D_i.num - D_i.size$$

- $\Phi(D_0) = 0$ ✓
- $\Phi(D_i) \geq 0 \quad \forall i$ ✓

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

<u>Case</u> insertion without expansion

$$c_i = 1$$

$D_i.num = D_{i-1}.num + 1$

$$\hat{c}_i = 1 + 2D_i.num - \cancel{D_i.size} - 2(D_{i-1}.num) + \cancel{D_{i-1}.size}$$

same

$\Rightarrow \quad \hat{c}_i = 3$ ✓

Case

insertion along an expansion

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

$$= D_i.\text{num} + 2 \cdot D_i.\text{num} - D_i.\text{size} \quad = -D_{i-1}.\text{size}$$

$$\overset{2}{=} \big(- 2 \cdot D_{i-1}.\text{num} + D_{i-1}.\text{size}$$

$$- D_i.\text{size} + D_{i-1}.\text{size} \qquad + 2 \cdot D_{i-1}.\text{size} = D_i.\text{size}$$

$$= -D_{i-1}.\text{size}$$

$$= D_i.\text{num} + 2 - D_{i-1}.\text{size}$$

$$= 3 \quad \checkmark$$

because This is an expansion step

$$D_i.\text{num} = D_{i-1}.\text{size} + 1$$

$\hat{c}_i = 3$ in both

cases

~~$\Rightarrow$ series of~~

$$\Rightarrow \sum c_i \leq \sum \hat{c}_i$$

$$= 3n \quad \checkmark$$

adding deletion

Def **load factor** $\alpha(T) :=$

$$\frac{\#\text{ of elts in } T}{\#\text{ of slots in } T}$$

in our series of expansions, the load factor varied between $1/2$ & 1

Table-deletion(T, x)

contraction

~~if $T.num = \frac{T.size}{2}$~~ $T.size - 1$

allocate a new-table w/ ~~$\frac{T.size}{2}$ slots~~ $\frac{T.size}{2}$

insert all elts of T into new-table

free T.table

T.table = new-table

$T.size = \frac{T.size}{2}$

delete x from T

T.num - -

Do $n/2$ insertions

do $n/2$ deletions

Doing $O(i)$ work

$i = n/2, \ldots, 1$

$= \Theta\left(\sum_{i=1}^{n/2} i\right) = O(n^2)$

problem: $n = 2^k$ do $n/2 + 1$ inserts - length of table $= n$

do a deletion, deletion, insertion, insertion, deletion, deletion, insertion, insertion

trigger a contraction $O(n)$

triggers an expansion $O(n)$

$n/4 \cdot \Theta(n) = \Theta(n^2)$

Tabel-deletion (T, x)

if $T.num \le \frac{T.size}{4}$ \

allocate new table ~~of~~ w/ T.num slots

insert all elts of T into new-table

free T. table

T. table = new-table

T. size = T. num

delete x from T

T. size --

what is the load factor here

$$1/4 \le \alpha(T) \le 1$$

We want to show a series of n table insertions + deletions, starting from empty table has O(n) operations.

use potential function method

$$\Phi(D_i) = \begin{cases} 2 \cdot D_i.num - D_i.size & \alpha \ge 1/2 \\ D_i.size/2 - D_i.num & \alpha < 1/2 \end{cases}$$

Same as before when $\alpha \ge 1/2$

ie we won't do any contaction in this case, ie the only expensive

expensive operation could be an expansion

why don't we use

$$\Phi(D_i) = 2 \cdot D_i.num - D_i.size$$

for all possible values of α?

when $\alpha < 1/2$, $2 D_i.num - D_i.size$ is negative

$\frac{D_i.size}{2} - D_i.num$ is ≥ 0 if $\alpha < 1/2$

(we've just proven that $\Phi(D_i) \geq 0$ $\forall i$

starting w/ empty array $D_0 = \emptyset$, $\Phi(D_0) = 0$

we want to bound $\hat{c}_i$

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Case i^{th} operation is
- insertion
- deletion

Cases for $\Phi(D_i)$
- $\alpha < 1/2$
- $\alpha \geq 1/2$

<u>Insertion</u>

both $\alpha(D_i)$ & $\alpha(D_{i-1}) \geq 1/2$

Then analysis is exactly the same as before. $\hat{c}_i \leq 3$

what if both $\alpha(D_i) + \alpha(D_{i-1}) < 1/2$

in this case, the insertion cannot require an expansion $\Rightarrow$ $c_i = 1$

$$\hat{c}_i = 1 + \Phi(D_i) - \Phi(D_{i-1})$$

$$= 1 + \left(D_i.size/2 - D_i.num\right) - \left(\frac{D_{i-1}.size}{2} - D_{i-1}.num\right)$$

no expansion $\Rightarrow$ equality of these terms

-1

$$\leq 0$$

Following the question I made at the end of the class: I was trying to ask in which case the sum of two "consecutive" load factors can be strictly smaller than 1/2, but it cannot

$\alpha(D_{i-1}) < 1/2$ & $\alpha(D_i) \geq 1/2$

$$\hat{c}_i = \underset{= 1 \text{ because we can't expand}}{c_i} + \Phi(D_i) - \Phi(D_{i-1})$$

$$\hat{c}_i = 1 + \left(2 \cdot D_i.num - D_i.size\right) - \left(\frac{D_{i-1}.size}{2} - D_{i-1}.num\right)$$

$*$

$$~~= 1 + 2(D_i.num) + 1 - D_{i-1}.size - \frac{D_{i-1}.size}{2} + \underbrace{D_{i-1}.num}_{D_i.num - 1}~~$$

$\alpha(D_i) \sim 1/2$

$$~~= 1 + \underbrace{3 \cdot D_i.num}_{\sim 3/2 \, D_i.size} - 3/2 \, D_{i-1}.size~~$$

$$~~\leq 3~~$$

a similar series of inequalities

shows

$\hat{c}_i \le 3$ for a deletion

& we get that a series of n insertions & deletions has $O(n)$ complexity.

* $\hat{c}_i = 1 + 2(D_i.num) - D_i.size - \frac{D_{i-1}.size}{2} + D_{i-1}.num$

we have that

$D_i.num = D_{i-1}.num + 1$

$D_i.size = D_{i-1}.size$

$= 1 + 3D_{i-1}.num + 2 - \frac{3}{2} D_{i-1}.size$

$\le 3 + \frac{3}{2} D_{i-1}.size - \frac{3}{2} D_{i-1}.size$

$= 3.$