CS 32 — Discussion 1B

Week 9

Hash Tables

Heaps

Hash Tables

Efficiency of lookup / deletion / insertion is contingent on #elements per bucket being small (i.e a constant)

$\Longrightarrow O(1)$

Rehashing :

Keep track of the load factor = $\frac{\text{total \# elements}}{\text{\# buckets}}$

Once load factor exceeds threshold (say 5), then

rehash : - Create a new array with twice as many buckets

- rehash all elements from old hash table into new table

Problem : periodically an insertion will cost $O(n)$

Fix: incremental rehashing.

- Maintain a hash table H with B buckets
- When H exceeds max load factor
  → create a new hash table H' w/ 2B buckets
- every time we do an insertion (still O(1))
  → insert into H'
  → rehash 5 elements from H to H'
- when H becomes empty: destroy it

Note:

lookup / deletion: now have to look in H & H' to find the element.

	~~H~~	H'	H''
# buckets	~~B~~	2B	4B
# elements	~~0~~	4B	

max load factor	:	2
rehashes / insertion	:	1
# bucket multiplier	:	2