DSA [03/06/2024]

DYNAMIC SET ← CAN CHANGE OVER TIME

INSERT
DELETE
SEARCH

DICTIONARY ADT ABSTRACT DATA TYPE

$O(1)$ EXPECTED
$O(n)$ WORST CASE

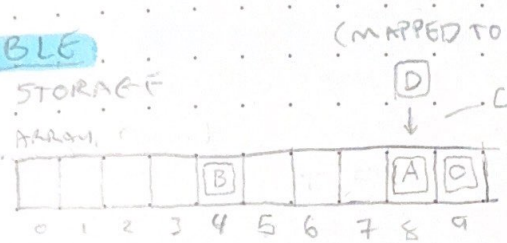
BASIC IDEA

INDEX = hashFn(VALUE TO BE STORED)

↑ GOAL TO MINIMIZE COLLISIONS AND FIND RESOLUTION

HASH TABLE

BACKING STORAGE → ARRAY



DIRECT MAP / ADDRESSING
(INT IN RANGE $0 \to n$)

EX

HASHTABLE SIZE m (IN THIS CASE $n=m$)
INSERT n ITEMS

*ALL HASH W/O COLLISIONS → UNIFORM HASH (OUR GOAL)

NORMALLY, WE HAVE COLLISIONS ∴

COLLISION RESOLUTION

1. CHAINING → MAKE AN ARRAY OF LISTS (SHOULD BE SHORT AND SPREAD OUT)
2. OPEN ADDRESSING → PICK ANOTHER SPOT

DSA [03/06/24]

HASHTABLE CHAINING

```
INSERT(x)
TABLE[getHash(x)].INSERT(x);
```

DUPLICATES? - SEARCH FIRST (IN GENERAL)
* FOR OUR ASSIGNMENT, DO NOT TEST FOR DUPLICATES

3 STYLES OF HASHING FUNCTIONS

m - TABLE SIZE
n - DATA SIZE

① $hash(K) = K \% M$ ← DIVISION METHOD (SHITTY)
IF USED USE A PRIME NUMBER FOR SIZE
(CONSTRAINS BEST TABLE SIZE SELECTION)

② $hash(K) = [m * (K*C - Floor(K*C)]$ ← C CONSTANT
m: TABLE SIZE
$K*C - Floor(K*C)$: LIKE A % LOCATION IN TABLE

MULTIPLICATION METHOD (GOOD) (WHAT WE USE)

③ UNIVERSAL HASHING
→ PICK CONSTANT C AT RUN TIME

HOW TO CHOOSE A CONSTANT C

→ IMPORTANT
→ INDEPENDENT OF SIZE
→ CHOOSE A C THAT MINIMIZES COLLISIONS
→ PICK A GOOD VALUE THAT IS LIKELY TO WORK WELL WITH UNKNOWN DATA
→ RECOMMENDATIONS: $\frac{1}{\Phi} \approx 0.61.8034$

GOLDEN RATIO: $\frac{\Phi + 1 = \Phi^2}{\Phi = 1.618...}$ ↖ $(\sqrt{5}-1)/2$

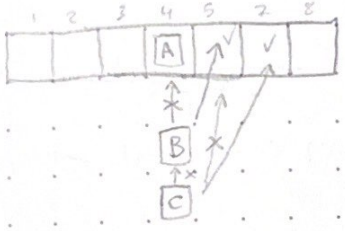
DSA [03/06/24]

OPEN ADDRESSING

METHODS:

① LINEAR PROBE → MOVE TO NEXT SLOT

→ $h'(k,i) = h(k) + i \% m$



(ALSO SHITTY)

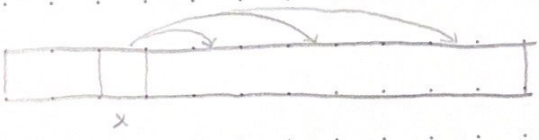
CAUSES PRIMARY CLUSTERING (LIKE A TRAFFIC JAM)

② QUADRATIC PROBE

→ $h'(k,i) = h(k) + i * C_1 + i^2 C_2 \% m$

→ WHERE C_1 AND C_2 ARE NON-0 CONSTANTS

→ CHOICE IS IMPORTANT (RELATIVELY PRIME)



"ACCELERATES ELEMENTS" (LOTS OF SMALL TRAFFIC JAMS)

"SECONDARY CLUSTERING" (NOT HORRIBLE, BUT DON'T USE)

③ DOUBLE HASHING

→ $h'(k,i) = h(k) + i * h_2(k) \% m$ ← PREVENTS PRIMARY AND SECONDARY CLUSTERING

$h(A) = n$

$h(B) = n$

$h(C) = n$

BEST SOLUTION!