CPSC 320
2010/03/25

CLRS
CHAPTER 17
AMORTIZED
ANALYSIS

BINARY COUNTER

EACH BIT FLIP COSTS 1 TIME UNIT

$$\ldots 000000$$

$+1$ (increment)

$$\ldots 00000\boxed{1} \qquad \text{cost} = 1 \text{ TU}$$

$+1$ (increment)

$$\ldots 0000\boxed{1}\boxed{0} \qquad \text{cost} = 2 \text{ TU}$$

$+1$

$$\ldots 00001\boxed{1} \qquad \text{cost} = 1 \text{ TU}$$

$+1$

$$\ldots 000\boxed{1}\boxed{0}\boxed{0} \qquad \text{cost} = 3 \text{TU}$$

$$\begin{array}{ccccccc} & 5\,4 & 3 & 2 & 1 & 0 \\ \ldots & 0\,0 & 0 & 1 & 0 & \boxed{1} \end{array} \qquad \text{cost} = 1 \text{TU}$$

ONE OPERATION OF n CAN COST $O(\lg n)$

THERE ARE n OPERATIONS.

SO, COST OF SEQ $\in O(n \lg n)$

MAYBE WE CAN FIND A TIGHTER BOUND.

TECHNIQUE #1: AGGREGATE ANALYSIS

COLUMN #0 ALWAYS CHANGES

$$\text{COST}(\text{COL}0) = n = \left\lfloor \frac{n}{2^0} \right\rfloor$$

COLUMN #1 CHANGES HALF THE TIME

$$\text{COST}(\text{COL}1) = \left\lfloor \frac{n}{2} \right\rfloor = \left\lfloor \frac{n}{2^1} \right\rfloor$$

$$\text{COST}(\text{COL}2) = \left\lfloor \frac{n}{4} \right\rfloor = \left\lfloor \frac{n}{2^2} \right\rfloor$$

$$\vdots$$

$$\text{COST}(\lfloor \lg n \rfloor) \approx \frac{n}{n} \approx 1 = \left\lfloor \frac{n}{2^i} \right\rfloor \quad \text{where } 1 \le \frac{n}{2^i} < 2$$

$$= \left\lfloor \frac{n}{2^{\lfloor \lg n \rfloor}} \right\rfloor$$

if $n = 2^i$

$\lfloor \lg n \rfloor = i$

$$\sum_i \text{COST}(i) = ?$$

$$\sum_{i=0}^{\lfloor \lg n \rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor \le \sum_{i=0}^{\lfloor \lg n \rfloor} \frac{n}{2^i}$$

$$= n \sum_{i=0}^{\lfloor \lg n \rfloor} \left(\frac{1}{2}\right)^i \le n \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i$$

$$= n \frac{1}{1-\frac{1}{2}} = 2n$$

COST OF SEQ $\in O(n)$

AVG COST PER OPERATION $\in O(1)$

TECHNIQUE #2: ACCOUNTING

PLAN: CHARGE TOO MUCH FOR
CHEAP OPERATIONS AND
"PAY FORWARD" FOR LATER
(EXPENSIVE) OPERATIONS

NO DEFICIT SPENDING

SO WE WILL END W/ $\geq$ \$0 ON
OUR STRUCTURE.

AND TAKING MONEY $\equiv$ OVERCHARGING OURSELVES

SO OUR TOTAL CHARGES = ACTUAL TIME + $\underbrace{\text{LEFTOVER CASH}}_{\geq 0}$
$\geq$ ACTUAL TIME

CHARGE \$2 per $0 \to 1$ flip
\$1 USED FOR FLIP
\$1 WE LEAVE ON THE 1

CHARGE \$0 per $1 \to 0$ flip
THE \$1 needed is ALREADY THERE.

COST OF AN INCREMENT?

COST(# of $0 \to 1$ flips PER INCREMENT = 1) = \$2

COST(# of $1 \to 0$ flips PER INCREMENT = ?) = \$0

TOTAL COST PER INCR = \$2 $\in O(1)$

TECHNIQUE #3: POTENTIAL METHOD

CREATE A POTENTIAL Φ AND USE IT LIKE OUR "CASH" IN ACCOUNTING.

"POTENTIAL" AS IN "POTENTIAL ENERGY"

AMORT TIME$_i$ = ACTUAL TIME$_i$ + $\Delta\Phi_i$

$$\sum_i \text{AMORT TIME}_i = \underbrace{\sum_i \text{ACTUAL TIME}_i}_{\text{TOTAL ACTUAL TIME}} + \underbrace{\sum_i \Delta\phi_i}_{\text{OVERALL } \phi \text{ CHANGE } \geq 0}$$

$\geq$ TOTAL ACTUAL TIME

$\Phi_0 = 0$

$\phi_i \geq 0$

$\phi(n) =$ ~~?~~ # of 1 bits in n

FOR A GIVEN INCREMENT i

* THE ACTUAL COST OF OP i is c_i
* THE $\Delta\Phi$ OF OP$_i$ is $\Delta\phi_i \overset{?}{=} 2 - c_i$

NOTE $c_i = \underbrace{\text{\# of } 1 \to 0 \text{ flips}}_{\uparrow \phi \text{ goes down here}} + \underbrace{1}_{\phi \text{ goes up here.}}$

$\phi' = \phi - (c_i - 1) + 1$

AMORT COST OF OP$_i$ = $c_i + \Delta\phi_i = c_i + 2 - c_i = 2$

CPSC 320
2010/03/30

~PROBLEM: STORE AN ARB # OF ELTS IN AN ARRAY

TARGET PER ~~INSERT~~ APPEND: $O(1)$

A: length
list of length slots
num: # of actual items stored

```
APPEND(A, x)
  if A.length == A.num:
    newA = new array[A.length * 2]

    for i = 1 to A.num
      newA[i] = A[i]
    newA.num = A.num
    // newA has room for at least one item
    A = newA
  A[A.num+1] = x
  A.num = A.num+1
```

costs A.num time units

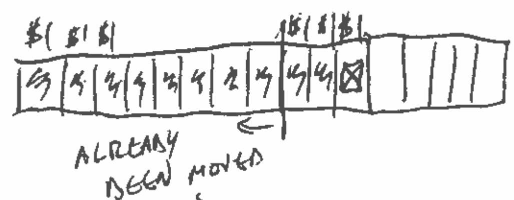
COST: 1 TIME UNIT

ONE APPEND IN A SERIES OF n CAN TAKE $O(n)$

EACH TIME WE DO THIS CHARGE \$3

* \$1 FOR ACTUAL COST
* ~~\$2~~ \$1 GOES ON THE NEW ELT TO PAY ITS FUTURE MOVE COST
* \$1 GOES TO PAY SOME OTHER ITEM'S MOVE COST.

IMAGINE



SO: "CHEAP" APPENDS GET CHARGED \$3
"EXPENSIVE" APPENDS ALSO GET CHARGED \$3

AMORT COST OF ONE APPEND IS $O(1)$

SPLAY TREE ANALYSIS

LEMMA: if $a+b \le c$, $a, b \in \mathbb{Z}^+$
$\lg a + \lg b \le 2\lg c - 2$

DEF'NS:

$S(\overset{\text{node}}{i})$ = "size" of i = # of descendants of i (which includes i itself)

$R(i)$ = "rank" of i = $\lg S(i)$

$\forall i, R(i) \ge 0.$

$\phi = \sum_i R(i)$

NOTE: if T is the root, then $R(T) = \cancel{\lg n}\ \lg n$ (and tree has n nodes)

TO PROVE: ANY ACCESS HAS AMORT COST $\in O(\lg n)$

splaying node X to the root has AMORT COST

$$\le 3(\underbrace{R(T)}_{\lg n} - \underbrace{R(X)}_{\ge 0}) + 1 \le 3\lg n + 1 \in O(\lg n)$$

How? ~~SHOW: ZIG COSTS ≤ 3(R~~

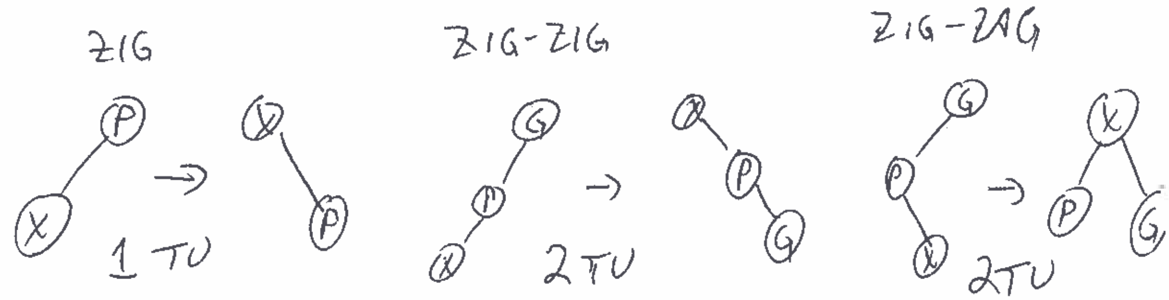
CPSC 320
2010/03/30

DEF'N ON A ZIG, ZIG-ZIG, OR ZIG-ZAG

$R_i(x)$ = initial rank of X

$R_f(X)$ = final rank of X

RECALL



SHOW AMORT COST OF ZIG $\leq 3(R_f(X) - R_i(X)) + 1$

" " OF ZIG-ZIG OR ZIG-ZAG $\leq 3(R_f(X) - R_i(X))$

IF WE ADD THESE UP ALONG THE SPLAY PATH, TERMS CANCEL TO

$$3(\underset{\text{last}}{\underline{R_f}}(X) - \underset{\text{first}}{\underline{R_i}}(X)) + 1 =$$

$$3(R(T) - R(X)) + 1$$

~~AMORT COST~~ $\cancel{\Delta\phi}$ OF ZIG?

AMORT COST

ACTUAL + $\Delta\phi$

$$1 + \cancel{R_f(X)} + R_f(p) - R_i(X) - \cancel{R_i(p)}$$

b/c $R_f(X) = R_i(p)$

$$= 1 + R_f(p) - R_i(X)$$

b/c $R_f(p) \leq R_f(X)$

$$\leq 1 + \underline{R_f(X) - R_i(X)}$$

b/c $R_i(X) \leq R_f($

$$\leq 1 + 3(R_f(X) - R_i(X)) \quad \checkmark$$

AMORT COST OF ZIG-ZIG $\quad$ b/c $R_f(X) = R_i(G)$

$$= 2 + \cancel{R_f(X)} + R_f(P) + R_f(G) - R_i(X) - R_i(P) - \cancel{R_i(G)}$$

$$= 2 + \underline{R_f(P)} + \underline{R_f(G)} - R_i(X) - \underline{R_i(P)}$$

b/c $R_f(P) \le R_f(X)$

$$\le 2 + R_f(X) + \underline{R_f(G)} - R_i(X) - \underline{R_i(P)}$$

b/c $R_i(X) \le R_i(P)$

$$\le 2 + R_f(X) + R_f(G) - \underline{R_i(X) - R_i(X)}$$

NOTE: ~~$S_f(P) + S_f(G) \le S_f(X)$~~

$$S_i(X) + S_f(G) \le S_f(X)$$

$\Downarrow$ LEMMA

$$\underline{R_i(X)} + \underline{R_f(G)} \le \underline{2R_f(X) - 2 - R_i(X)}$$

$$\le \cancel{2} + R_f(X) + 2R_f(X) - \cancel{2} - R_i(X) - 2R_i(X)$$

$$\le 3R_f(X) - 3R_i(X)$$

✓ ☺

OUT-OF-CLASS: ZIG-ZAG CASE $\quad$ b/c $R_f(X) = R_i(G)$

AMORT COST OF ZIG-ZAG $= 2 + \cancel{R_f(X)} + R_f(P) + R_f(G) - R_i(X) - R_i(P) - \cancel{R_i(G)}$

$$= 2 + R_f(P) + R_f(G) - R_i(X) - R_i(P)$$

b/c $R_i(X) \le R_i(P)$

$$\le 2 + R_f(P) + R_f(G) - R_i(X) - R_i(X)$$

$$= 2 + R_f(P) + R_f(G) - 2R_i(X)$$

$$\le 2 + 2R_f(X) - 2 ~~R_f(G)~~ - 2R_i(X)$$

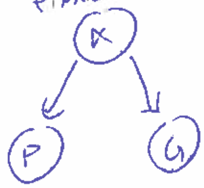
$$= 2(R_f(X) - R_i(X))$$

b/c $R_f(X) \ge R_i(X)$

$$\le 3(R_f(X) - R_i(X))$$

☺ ✓

Note:
FINAL CONFIG



SO; $S_f(X) \ge S_f(P) + S_f(G)$

$\therefore 2R_f(X) - 2 \ge R_f(P) + R_f(G)$ (by LEMMA)

SO: SUM OF AMORT COSTS OF A SPLAY $\le 3(R(T) - R(X)) + 1 \in O(\lg n)$