CSL 356 Lecture 11, Aug 20

Continuing with semi-dynamic (only insertions) dictionary

Given a set of elements S with $|S| = n$, we maintain sorted arrays according to the binary representation of n. (Almost $\log n$ arrays)

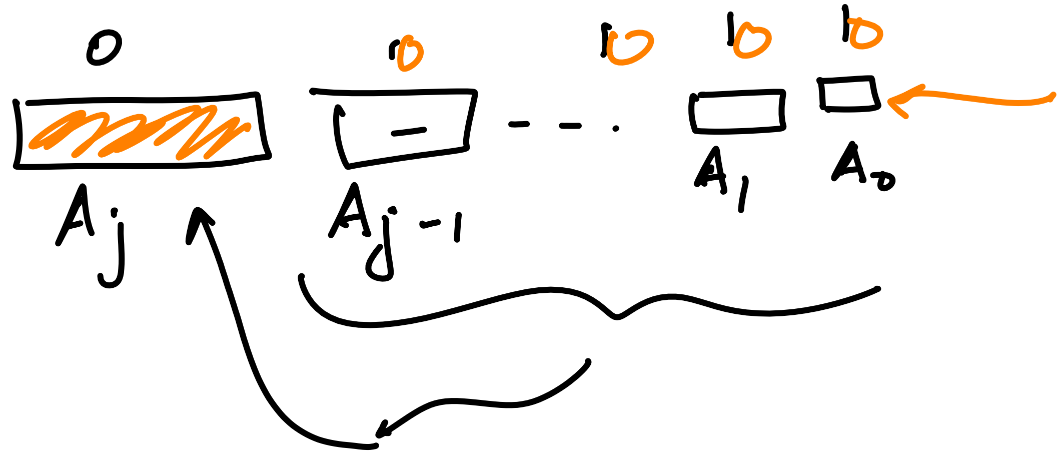
Search : $O(\log n \cdot \log n) = O(\log^2 n)$

Insertion : Create a new array A_j with $|A_j| = 2^j$ given that all the arrays $A_0, A_1 \ldots A_{j-1}$ were full when the insertion occurred

A_j is created by combining all the elements from $A_0 \ldots A_{j-1}$ (about $2^j - 1 + 1 = 2^j$ elements)

Can be done in $O(2^j)$ comparisons

2^j can be large (upto n)



How often do we pay the price for filling up A_j ?

Observation : Only once after every 2^j insertions, A_j is affected.

$\Rightarrow$ Over a sequence of m insertions the number of times, we incur a cost for A_j $= O\left(\frac{m}{2^j}\right)$

$\Rightarrow$ Total cost $= \sum_j O\left(\frac{m}{2^j} \times 2^j\right) = O(m \log m)$

So the "amortised" cost of insertion is $O(\log m)$ (amortised over a "large" no operations)

Goal is to get a worst case bound over a <u>sequence of operations</u> and look at the amortised cost

<u>Counter</u> can count to N

$b_{\log n}$ ... b_1 b_0

0 0 0 0 0 0

modulo N counter

. 0 1 0 1

— - 0 1 1 0

1 1 1 1 1 1 1 1 1

0 0 0 1 0 0 0 0 0

Worst case cost of a single operation say T

Then worst case cost of m operations is $O(T \cdot m)$

$T = \log N \qquad m = N \qquad O(N \log N)$

Total # of bits flipped

$$= \sum_j \frac{N}{2^j} \cdot (j \cdot) = N \cdot \sum_j \frac{j}{2^j} = O(N)$$

$$\begin{array}{cccccccc} & & & 5 & 4 & 3 & 2 & 1 \\ - & \cdot & 1 & 0 & 1 & 1 & 1 & 1 \\ & & & 1 & 0 & 0 & 0 & 0 \end{array}$$

Amortised Analysis

Potential based amortised analysis

Associate a "potential function" $\phi(\,)$ with the state of the data structure. $\phi(i)$ is the potential at state i

Amortised cost at step i

= change of potential + actual cost

$= \phi(i+1) - \phi(i) + w_i$

Total amortised cost $= \sum_i$ Amortised cost in step i

$$= \sum_i \phi(i+1) - \phi(i) + \sum_i w_i$$

$$= \phi_f - \phi_{initial} + W$$

final potential

initial potential

total actual cost

Total Amortised cost $= \phi_f - \phi_I + W$

$\Rightarrow W =$ Total amortised cost $-(\phi_f - \phi_I)$

$W \leq$ Total amortised cost if $\phi_f - \phi_I$ is non-negative

For the counter example what is a good potential function

# of bits $= 1$

Amortised cost of incrementing the counter

$$1\,0\,1\,1\,0\,\underbrace{1\,1\,1} \rightarrow 1\,0\,1\,1\,1\,\underbrace{0\,0\,0}$$

Potential change $= -3 + 1$

Amortized cost : $O(1)$

Over a sequence of N

increments T.! Amortized cost : $O(N)$