15-210 : Oct 11, 2017

Today: Union / Merge in $O(\log(n+m))$ span
(Will assume keys are unique so Union = Merge)

Reminder: in class showed that

Union takes $O(\log n \log m)$ span

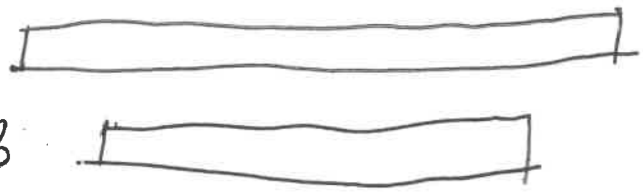
```
union (A, B) =
    case (A, B) of
        (leaf, _) => B
        (_, leaf) => A
        (Node(A_L, k, A_R), B) =>                    log m span
          let (B_L, _, B_R) = split (A, k)
              (L, R) = union(A_L, B_L) || union(A_R, B_R)
          in joinMid (L, K, R)
                     log(n) span
```

Dual Binary Search

or kthSmallest on two sorted Sequences (Sets, Tables)

A

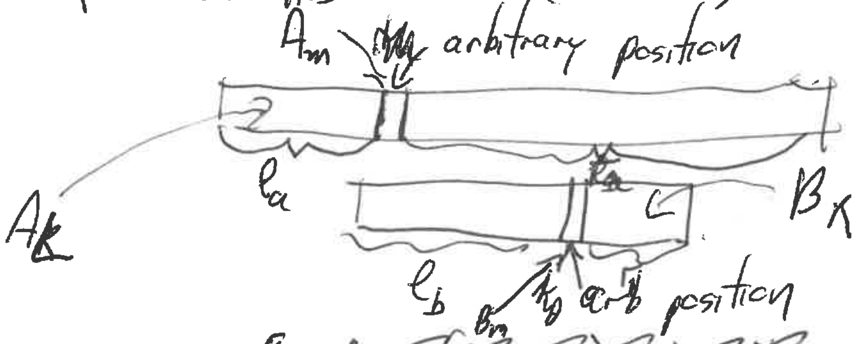
B



sorted

goal:

k^{th} smalles of $(A \cup B)$



~~if~~ $\cancel{k \le (l_a + l_b) + 1}$

If $(A_m < B_m)$ then there are $(l_a + l_b + 1)$
elements less than B_m and $(r_a + r_b + 1)$
elements greater than A_m

so if $k \le (l_a + l_b + 1)$ can toss $B_r^{+B_m}$ since $(B_R > B_m)$

otherwise can toss A_L $(A_L < A_m)$

---

type of ~~Typical~~ question on google interviews assume size(T) constant work

(appologize if off by one errors)

```
KthSmallest (A, B, K)
  case (expose(A), expose(B)) of
    (Leaf, _) => Nth (B, k)
    (_, Leaf) => Nth(A, k)
    (Node(A_L, A_m, A_R), Node (B_L, B_m, B_R)) =>
    let l = size(A_L)+size(B_L)+1
    in
      if
      case (A_m < B_m, K ≤ l) of
        (true, true) => KthSmallest (A, B_L, K)
        | (true, false) =>    ||    (A_R, B, K-size(A_L)+1)
        | (false, true) =>    ||    (A_L, B, k)
                                    (A, B_R, K-size(B_L)+1)
```

Span = Depth(A) + Depth(B)

= log(n+m) if balanced trees

Can imp union using kthSmallest

Union(A, B)

case (expose(A), expose(B)) of

Base cases

(_, _) ⇒ let ~~m~~ m = ~~split~~ kthSmallest(A, B, $\frac{|A|+|B|}{2}$)

~~(L, R) = split(~~

$(A_L, \_, A_R) = split(A, m)$

$(B_L, \_, B_R) = split(B, m)$

$(L, R) = union(A_L, B_L) \,||\, union(A_R, B_R)$

in joinMid(~~L~~, m, R) end

No improvement, but at least now perfect balance on recursive calls

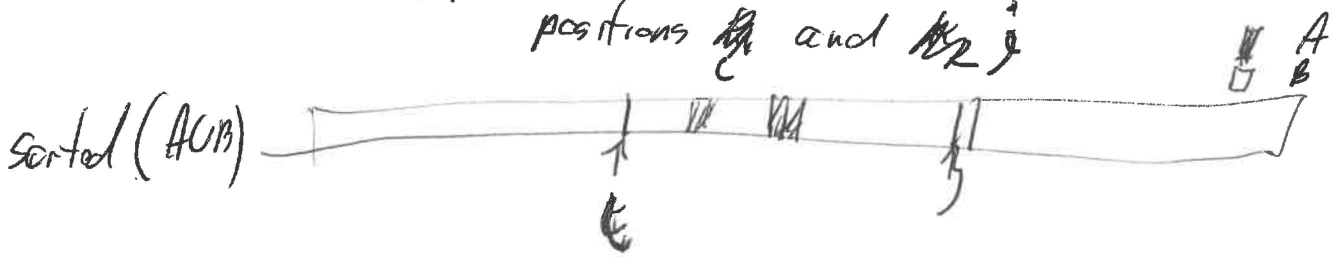
Idea to reduce span

— break up more quickly (i.e. into more smaller pieces)

Lets try $\sqrt{n}$ pieces

$\text{Sub}(A, B, i, j) \rightarrow (A_s, B_s)$

extracts elements between ~~k_1 and k_2~~ positions i and j



let $m_1 = \text{kthSmall}(A, B, i)$

$m_2 = \text{kthSmall}(A, B, j)$

$(A_L, \_, \_) = \text{split}(A, m_2)$

$(\_, \_, A_s) = \text{split}(A_L, m_1)$

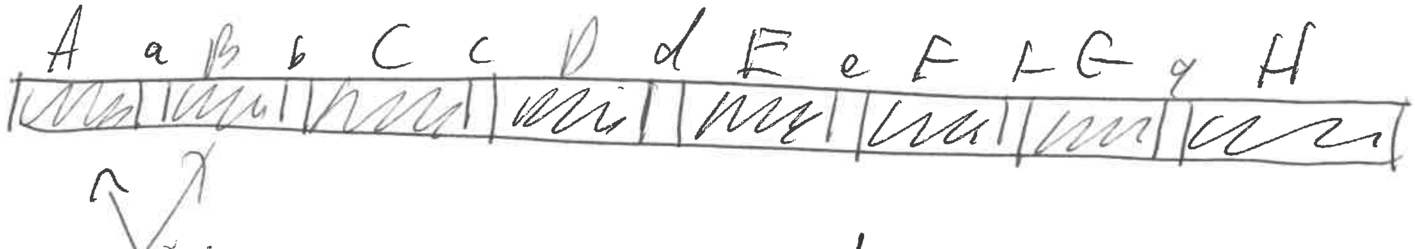
same for B_s

Break into $\sqrt{n}$ parts, union each

$n = |A| + |B|$

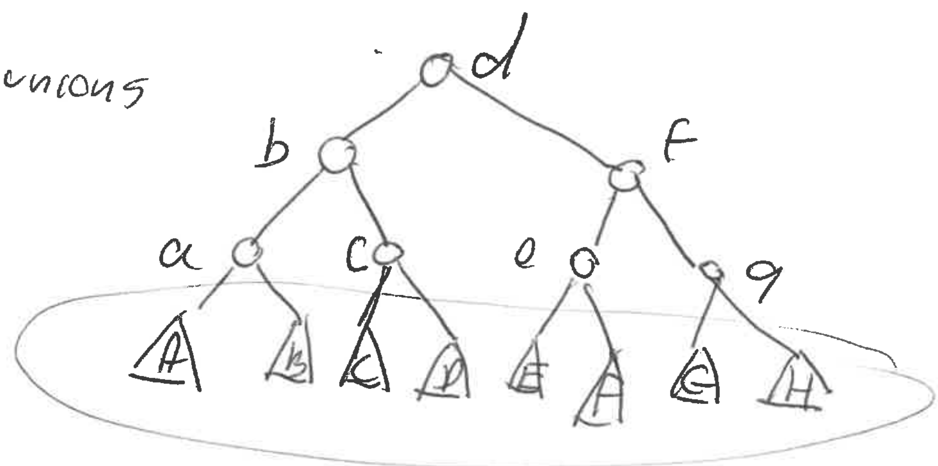
$$\langle \text{union}(\text{sub}(A, B, i\sqrt{n}, (i+1)\sqrt{n})) : 0 \le i \le \sqrt{n} \rangle$$

need to deal with rounding



results of unions

all balanced and same size



easy to construct in $\log\sqrt{n}$ span

Overall Span: ~~(Assume equal size subs)~~

$$S(n) = S(\sqrt{n}) + \cancel{(\log n + \log n)}\ \log n + \log\sqrt{n}$$

$$\le S(\sqrt{n}) + k\log n \quad \text{root dominated} \quad = O(\log n)$$

$k \log n$

$k(\log\sqrt{n}) = \frac{k}{2}\log n$

Tricky part: showing work is

$O\left(m \log\left(\frac{n+m}{m}\right)\right)$

not hard if assume split evenly

a little tricky for general case

One other way to do it when $n \approx m$ (same size)

$S = (\log n)$ 1) split into $\frac{n}{\log n}$ pieces of size $\log n$ each using sub

$S = (\log n)$ 2) sequentially ~~union~~ merge/union each piece

$S = (\log n)$ 3) put results together