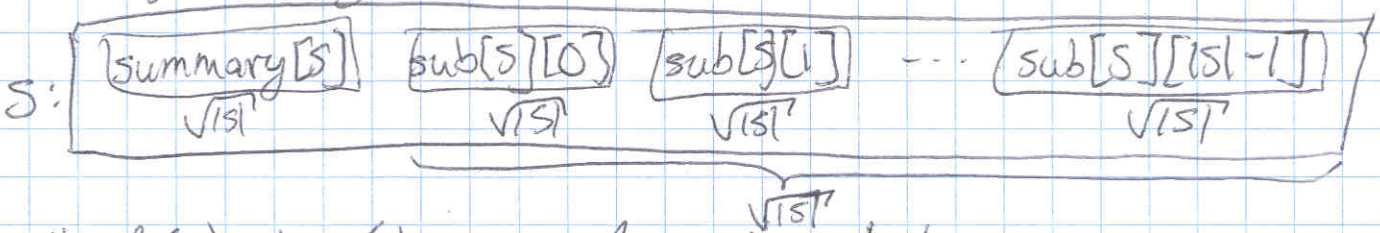
6.897
L1.3
Feb. 10, 2003

③ Recursion

- overall structure of size u.
- recursively split each structure S of size $|S|$ into $\sqrt{|S|}$ substructures each size $\sqrt{|S|}$.
  $sub[S][0]$, $sub[S][1]$, ..., $sub[S][|S|-1]$.
- store summary structure $summary[S]$ of size $\sqrt{|S|}$ representing which substructures are nonempty.



- high(x), low(x) now depend on $|S|$

```
Insert(x, S):
    if sub[S][high(x)] empty: Insert(high(x), summary[x])
    Insert(low(x), sub[S][high(x)])
```

Time: $T(u) = 2T(\sqrt{u}) + O(1)$
$T'(u) = 2T'(\frac{1}{2}\lg u) + O(1) = O(\lg u)$!

```
Successor(x, S):
    j ← Successor(low(x), sub[S][high(x)])        } T(√u)
    if j < ∞: return high(x)√|S| + j
    else: i ← Successor(high(x), summary[S])      } T(√u)
          j ← Successor(-∞, sub[S][i])            } T(√u)
```

Time: $T(u) = 3T(\sqrt{u}) + O(1) = O((\lg u)^{\lg 3}) = \omega(\lg u)$. !

④ Maintain min & max of every recursive structure → Fix Successor

- $O(1)$ overhead in Insert

```
Successor(x, S)
    if low(x) < max[sub[S]][high(x)]:
        return high(x)√|S| + Successor(low(x), sub[S][high(x)])
    else: i ← Sucessor(high(x), summary[S])
        return i·√|S| + min[sub[S][i]]
```

$O(\lg \lg u)$