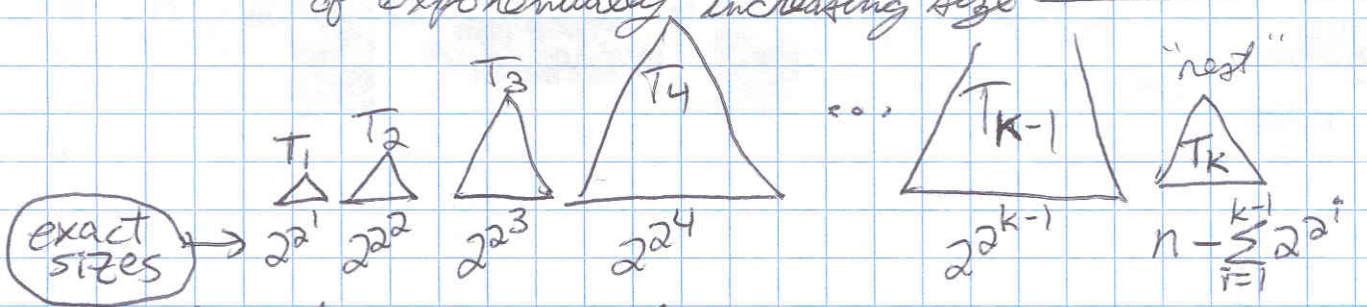
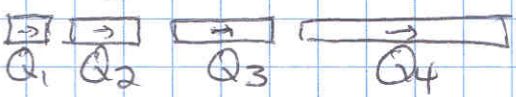
Working set structure (a warmup)

6.897
L8.2
Mar. 10, 2003

- lg lg n balanced binary search trees of exponentially increasing size

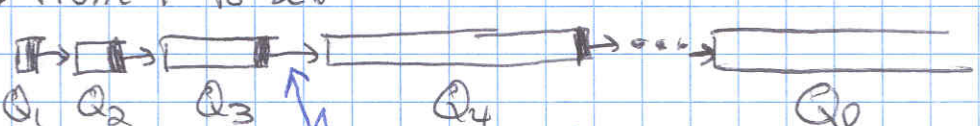


- queue Q_i storing same elements as T_i



- pointers between corresponding elts. in T_i & Q_i.

Search(x):
- look in T_1, T_2, ... until found
- unsuccessful cost: $\sum_{i=1}^{k} O(\lg 2^{2^i}) = \sum_{i=1}^{k} O(2^i) = O(2^k) = O(\lg n)$ fine.
- suppose found in T_ℓ
- delete x from T_ℓ and Q_ℓ
- insert x into T_1 and enqueue $\rightarrow Q_1$ } like MTF
- shift from 1 to ℓ.



(fixes sizes)

Dequeue(Q_i) $\rightarrow y$
Delete y from T_i
Insert y into T_{i+1}
Enqueue $y \rightarrow Q_{i+1}$

Cost: touch each tree $O(1)$ times
trees touched $= \ell \approx \lg \lg$ (position of x in "big queue")
position of x in big queue $=$ # distinct elements accessed since x last accessed $=: t$

$$\text{time} = \sum_{i=1}^{\ell} O(\lg 2^{2^i}) = O(2^\ell) = O(2^{\lg\lg t}) = O(\lg t)$$

Insert/Delete also possible using shift

Similar to OBR
$\rightarrow 2^{2^i}$ vs. 2^i