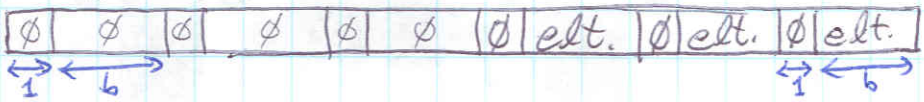
6.897
L13.3
Mar. 17, 2005

Packed sorting: $w \geq 2(b+1)\lg n \cdot \lg\lg n$ (for convenience)
$O(n)$ time

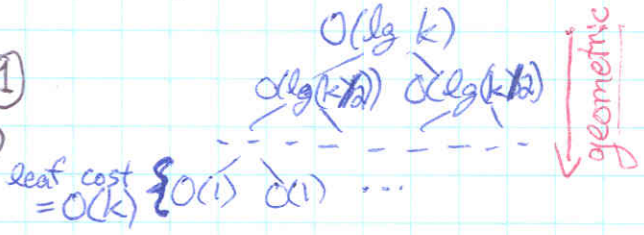
- pack $\lg n \lg\lg n$ elements into a word:

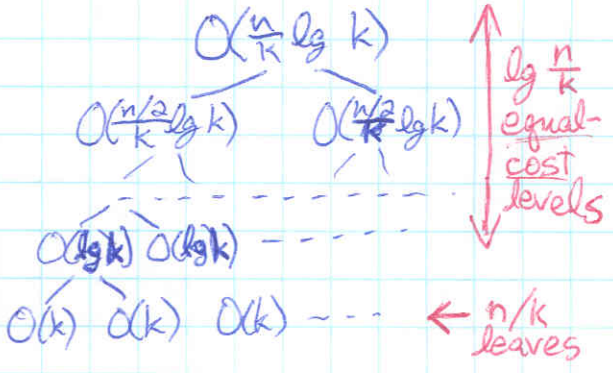


- room for twice as many elements (for merging)

① merge pair of sorted words with $k \leq \lg n \lg\lg n$ elements into one sorted word with $2k$ elements in $O(\lg k)$ time
- hardest step
- bitonic sort + bit tricks

② mergesort $k = \lg n \lg\lg n$ elements & pack into one word in $O(k)$ time
- binary tree of merges via ①
- $T(k) = 2T(k/2) + O(\lg k)$
  $= O(k)$



③ merge two sorted lists of r sorted words into one sorted list of $2r$ sorted words in $O(r \lg k)$ time
- like standard merge but with ① as comparator
- merge first word of each list → two words
- output first word
- put second word at front of list containing max(second word)

④ mergesort with ③ as merger & ② at base case
- $T(n) = 2T(n/2) + O(\frac{n}{k} \lg k)$ (③; $r = \frac{n}{k}$)
- $T(k) = O(k)$ (②)
- $\Rightarrow T(n) = O(\frac{n}{k} \lg k \lg \frac{n}{k} + \frac{n}{k} \cdot k)$
  $\leq O(\frac{n}{k} \lg k \lg n + n)$
- substitute $k = \lg n \cdot \lg\lg n$
- $\Rightarrow T(n) = O(\frac{n}{\lg n \lg\lg n} \lg\lg n \cdot \lg n + n)$
  $= O(n)$