

Lecture 14: Wednesday February 5, 2003

today

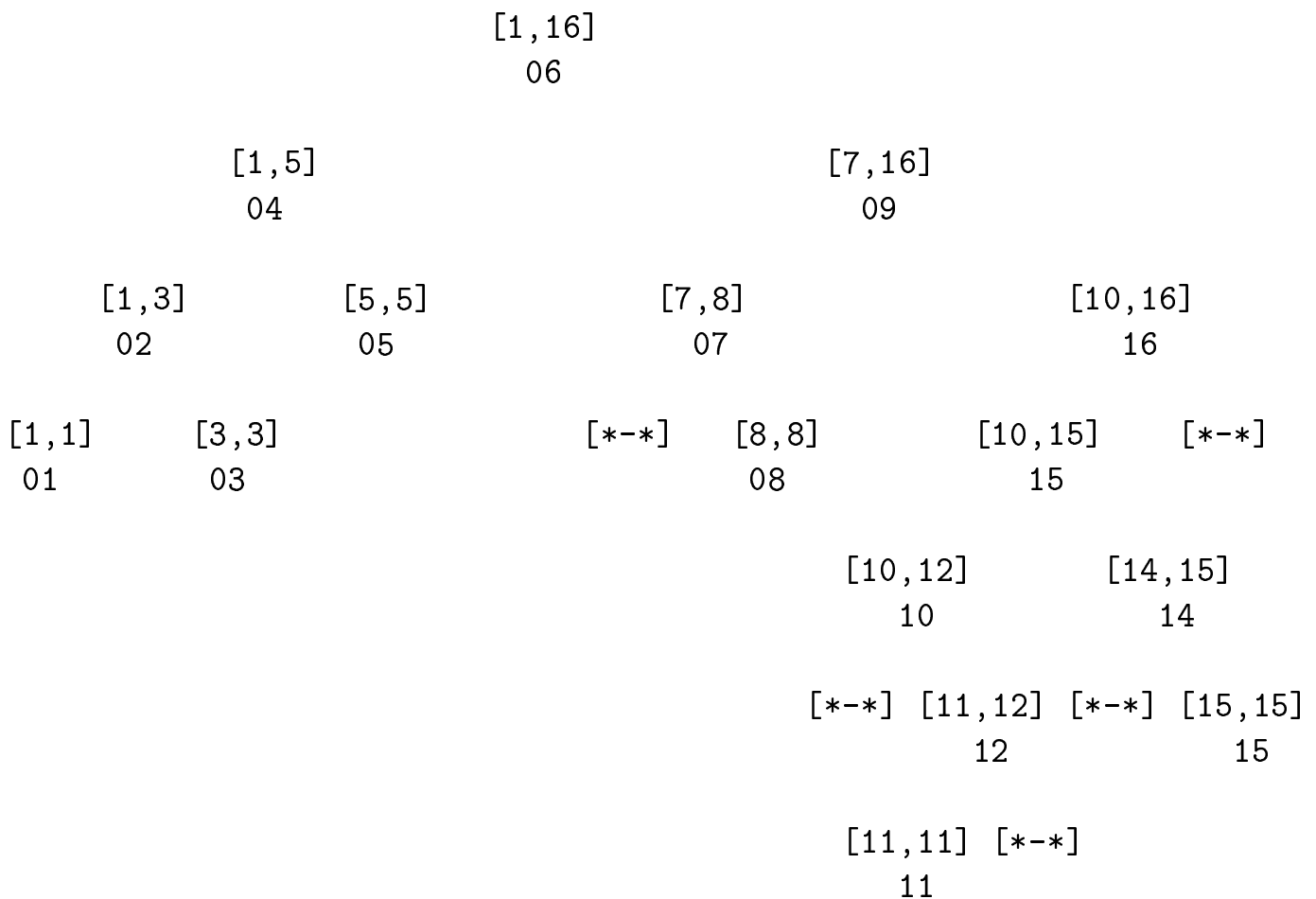
- QS analysis and recursion trees
- a bad case for QS (actually WC)
- a good case for QS (actually BC)

QS analysis and recursion trees

- QS run time proportional to number key comparisons why?
- QS recursion tree makes counting KC easy, because to count KC, only need know (at each call) rank of split key

a 16 key example: a 16 key QS recursion tree, showing rank of

- split key (if partition called)
- single key (if sublist has size 1)



- observe: in resulting tree, for each node, all keys(left subtree) \leq node's key \leq all keys(right subtree)
- 1-1 correspondence: quicksort recursion tree \leftrightarrow binary search tree

QS run time: a bad case

01

02

03

04

05

06

...

- number of KC in above case?

- $$W(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ n - 1 + W(0) + W(n - 1) & \text{if } n \geq 2 \end{cases}$$

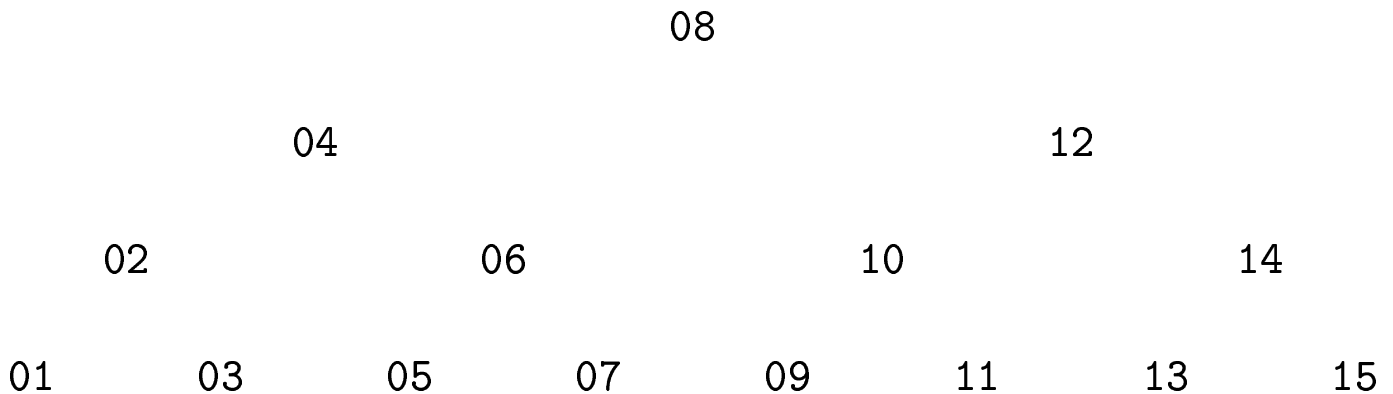
- $$W(n) = \sum_{j=n-1}^1 j = n(n-1)/2$$

- so QS WC KC $\geq n(n-1)/2$

QS run time: worst case

- previous example is a worst case, because
- QS WC KC $\leq n(n - 1)/2$:
 - each KC is between splitter/other
 - after partition, splitter never compared again
 - so each pair of elements compared ≤ 1 time
 - $\binom{n}{2} = n(n - 1)/2$
- conclusion: QS WC
 - KC = $n(n - 1)/2$
 - so run time in $\Theta(n^2)$

QS run time: a good case



- number of KC in above case?
- $B(1) = 0$
- $B(n) = n - 1 + B(\lfloor n - 1/2 \rfloor) + B(\lceil n - 1/2 \rceil)$ why?
- $B(n) \approx n + 2B(n/2)$
- can show $B(n) \in \Theta(n \log n)$ e.g. use master theorem
- conclusion: QS BC KC in $O(n \log n)$
- is QS BC KC in $\Theta(n \log n)$?
- yes: can show (e.g. by induction) $B(n)$ is best case number of KC, so
- QS best case run time in $\Theta(n \lg n)$