Max Array Problem

Note Title 9/11/2007

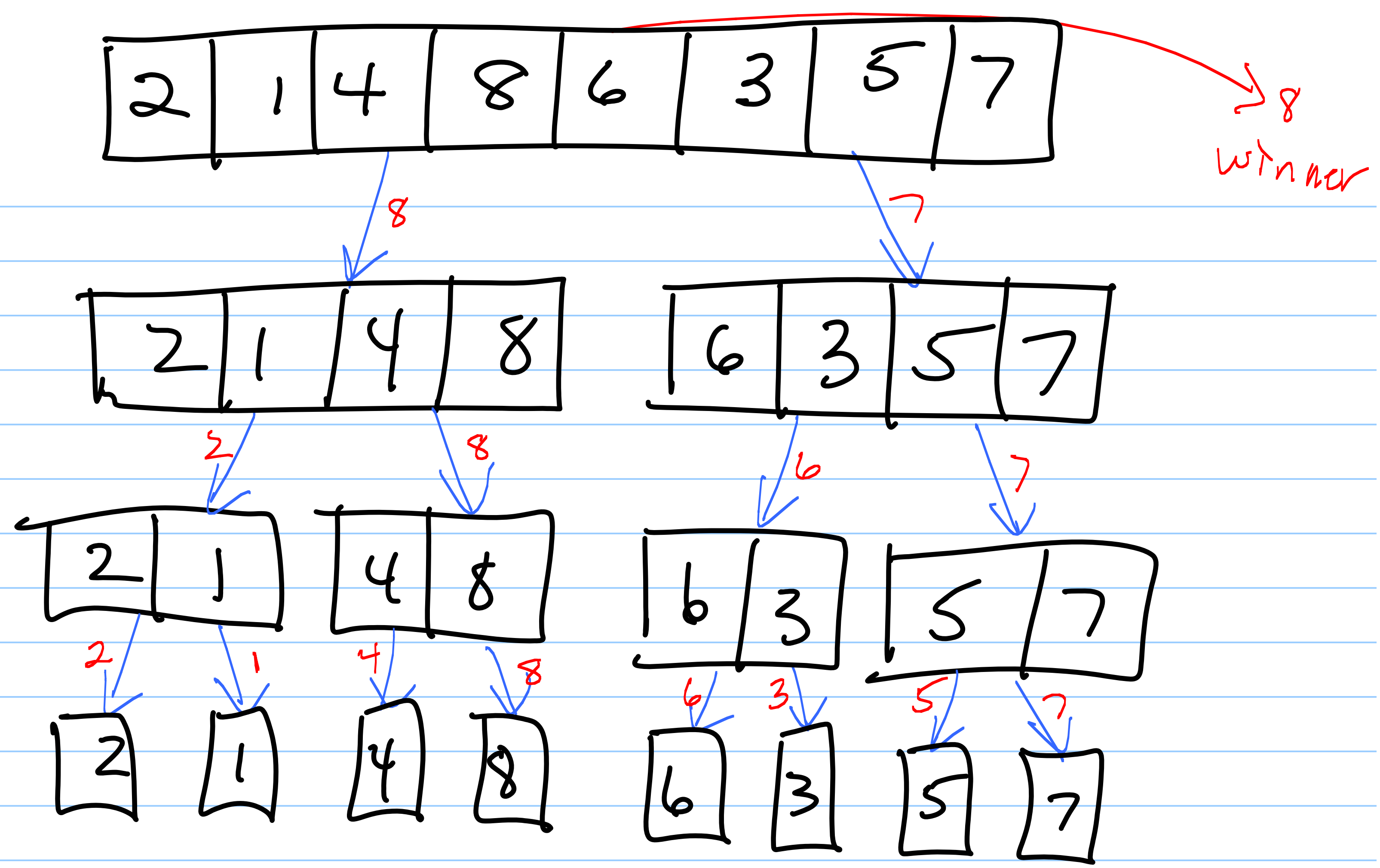
2 1 4 8 6 3 5 7 max value 8 by inspection

1) Sort array at cost Θ(nlogn). Take last item.
2) Do pair-wise comparisons through file.
   This implies n-1 comparisons, so cost is linear
3) Use recursive procedure - Find winner each half & compare to find next winner

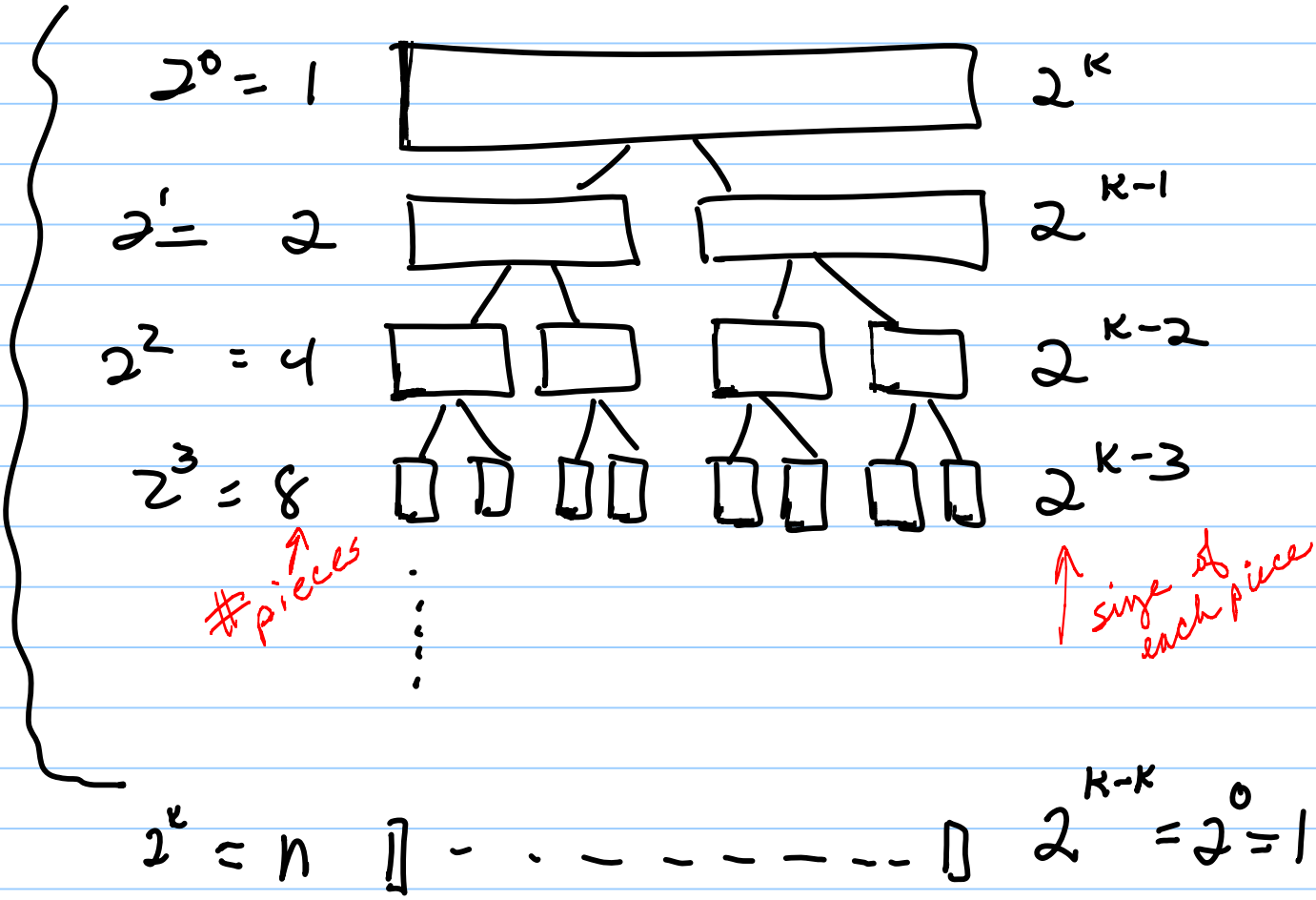
This is better

Looks expensive!

★ $\Theta(n \log n)$ is nominally the cost of all compaison based sorting.
See module 13.


2 1 4 8 6 3 5 7
8
winner
2 1 4 8
6 3 5 7
2 1
4 8
6 3
5 7
2
1
4
8
6
3
5
7

$n = 2^k$ - why can we assume this w/o a problem?



Cost is $\sum_{i=0}^{k-1} 2^i = \frac{2^k - 1}{2-1} = 2^k - 1$ (uses A.5 shown below)

A.5 says $\sum_{k=0}^{n} x^k = \frac{x^{n+1} - 1}{x - 1}$ $x = 2$ here

Note $n = 2^k$ so $2^k - 1$ is $n - 1 \Rightarrow \Theta(n)$