Sorting Algorithms and Master Theorem Summary 1. Bubble Sort Not a divide-and-conquer algorithm.
Complexity: $\Theta(n^2)$ time, $O(1)$ space. 2. Merge Sort Recurrence: $T(n) = 2T(n/2) + \Theta(n)$
Master Theorem ($a=2$, $b=2$, $f(n)=\Theta(n)$) $\Rightarrow$ Case 2 $\Rightarrow$ $\Theta(n \log n)$.
Space: $O(n)$. 3. Quick Sort (Random Pivot) Average: $T(n) = 2T(n/2) + \Theta(n) \Rightarrow \Theta(n \log n)$
Worst: $\Theta(n^2)$
Space: $O(\log n)$. 4. Quick Sort (Median-of-Three Pivot) Better pivot choice, fewer unbalanced cases.
Average: $\Theta(n \log n)$, Worst: $\Theta(n^2)$, Space: $O(\log n)$. 5. Heap Sort Build heap $O(n)$ + n×extract $O(\log n) = O(n \log n)$
Space: $O(1)$. Summary Table AlgorithmBestAverageWorstSpace Bubble Sort$O(n)O(n^2)O(n^2)O(1)$ Quick Sort (random)$O(n \log n)O(n \log n)O(n^2)O(\log n)$ Quick Sort (median-3)$O(n \log n)O(n \log n)O(n^2)O(\log n)$ Merge Sort$O(n \log n)O(n \log n)O(n \log n)O(n)$ Heap Sort$O(n \log n)O(n \log n)O(n \log n)O(1)$ **Conclusion:** According to the Master Theorem, Merge Sort and Quick Sort have $\Theta(n \log n)$ expected complexity, while Bubble Sort is much slower.