CheatSheet

Quick Notes

For loops = n, nested for loops = n^(number of for loops)
Divide and conquer tends towards $n \log n$ (merge sort)\

QuickSort - Relies on a good pivot for an efficient implementation

Merge sort is especially useful for linked lists

Stability is important for key-based systems. Specifically if key order is to be maintained. Ignore stability for integers for example.

Algorithm complexity

- Quick sort
  - BEST: $\Omega(n \log n)$
  - AVERAGE: $\ominus(n \log n)$
  - WORST: O(n^2)
- merge sort
  - BEST: $\Omega(n \log n)$
  - AVERAGE: $\ominus(n \log n)$
  - WORST: O($n \log n$)
- Selection sort
  - BEST: $\Omega(n^2)$
  - AVERAGE: $\ominus(n^2)$
  - WORST: O(n^2)
- Insertion sort
  - BEST : $\Omega(n)$ IFF your list is almost sorted already
  - AVERAGE: Same as selection
  - WORST: Same as selection
- Rabin-Karp
  - Match: $\ominus(m)$
  - FIND
    - AVERAGE: $\ominus(n+m)$
    - WORST: $\ominus((n-m)m)$
- Rabin-Morris-Pratt
  - MATCH: $\ominus(m)$
  - FIND
    - AVERAGE: $\ominus(n)$
- Boyer-Moore
  - MATCH: $\ominus(m+k)$
  - FIND
    - AVERAGE: (n/m)
    - WORST: (mn)