Algorithms Cheat Sheet

by alexk12 via cheatography.com/149245/cs/32446/

Complexities

Name	Worst case	Average case	Best case
Fractional Knapsack	O(n*log(n))		
0-1 Knapsack	O(n*W)		
MergeSort	O(n*log(n))		
Selection-Sort	O(n^2)		
Insertion-sort	O(n^2)		
Heap-sort	O(n*log(n))		
Quick-sort	O(n^2)		O(n*log(n))
Comparison-Based Sort			Ω(n*log(n))
BucketSort	O(n + N)		
RadixSort	O(d*(n + N))		
QuickSelect	O(n^2)	O(n)	O(n)

Divide-and-Conquer

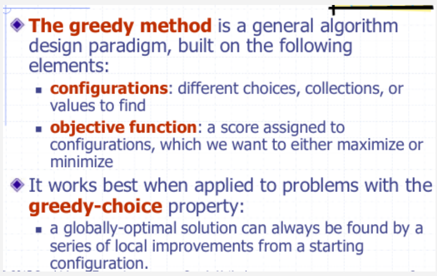
Divide-and conquer is a general algorithm design paradigm:

-> Divide: divide the input data S in two or more disjoint subsets S1, S2, ...

-> Conquer: solve the subproblems recursively

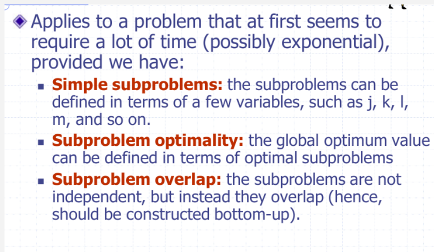
-> Combine: combine the solutions for S1, S2, ..., into a solution for S

Greedy Method



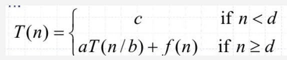
- **The greedy method** is a general algorithm design paradigm, built on the following elements:
  - **configurations**: different choices, collections, or values to find
  - **objective function**: a score assigned to configurations, which we want to either maximize or minimize
- It works best when applied to problems with the **greedy-choice** property:
  - a globally-optimal solution can always be found by a series of local improvements from a starting configuration.

Dynamic Programming



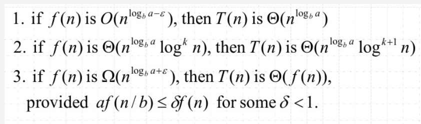
- Applies to a problem that at first seems to require a lot of time (possibly exponential), provided we have:
  - **Simple subproblems:** the subproblems can be defined in terms of a few variables, such as j, k, l, m, and so on.
  - **Subproblem optimality:** the global optimum value can be defined in terms of optimal subproblems
  - **Subproblem overlap:** the subproblems are not independent, but instead they overlap (hence, should be constructed bottom-up).

Master Theorem



$$T(n) = \begin{cases} c & \text{if } n < d \\ aT(n/b) + f(n) & \text{if } n \ge d \end{cases}$$

Master Theorem



1. if $f(n)$ is $O(n^{\log_b a - \varepsilon})$, then $T(n)$ is $\Theta(n^{\log_b a})$
2. if $f(n)$ is $\Theta(n^{\log_b a} \log^k n)$, then $T(n)$ is $\Theta(n^{\log_b a} \log^{k+1} n)$
3. if $f(n)$ is $\Omega(n^{\log_b a + \varepsilon})$, then $T(n)$ is $\Theta(f(n))$, provided $af(n/b) \le \delta f(n)$ for some $\delta < 1$.