Date: 18/02/19
Page No.

6 Recursive Patterns

1. Iteration
2. Breaking into subproblem
3. selection
4. Divide and conquer
5. Ordering
6. Depth first search

Iteration: Iterate over data list using recursion
rarely useful in simplifying code
example: print linked list, factorial, for loop

Breaking into subproblem:
classic subproblem, technically all recursive problem but many are not obvious hence the pattern
Use this pattern when it make sense
Example: fibonacci sequence,

selection (combination):-
fundamentally, problems that can be solved by finding all valid combination.
Brute force: find all and validate every combination
optimize by validating as we go / backtracking
Example: knapsack, N-queen
phone spell
N-Queens

Date :
Page No.

Ordering (permutation) :

Similar to selection except ~~those~~ ~~if~~ order matters.

Brute force - find all ~~combination~~ permutation and validate which is best / watch your condition.

Example

Find permutation of certain up to N-digit nos. whose digit sum is certain no.

Word squares.

Divide and Conquer :-

Classic recursive prob.

Can we solve the prob. for each half of the I/P and easily combine the result.

Common with searching and sorting.

Depth first search :-

Common technique for tree / graph structure

eg. Search in tree.

Probability of a [illegible]