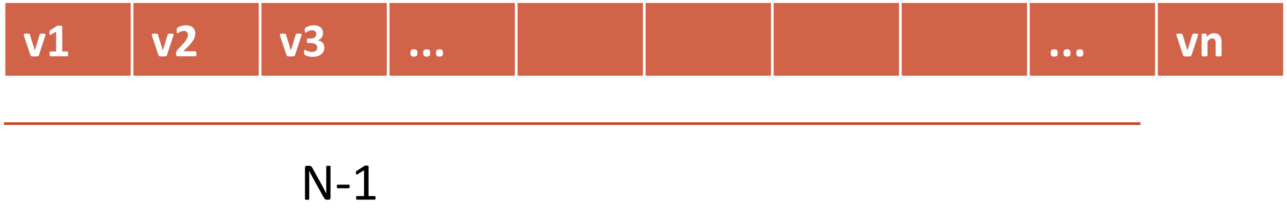
CS F364 Design & Analysis of Algorithms

Divide-And-Conquer : Structural Induction

STRUCTURAL INDUCTION – EXAMPLE 1-A

- List / Array: Induction on size (say, N)



Consider the sub-problem of size N-1

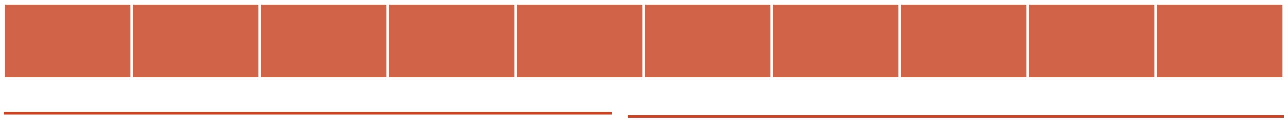
Where is this used? Provide an example.

Generalize this to a sub-problem of size N-k for constant k

Find an example with k>1.

STRUCTURAL INDUCTION - EXAMPLE 1-B

- List / Array: Induction on size (say, N)
  - Consider sub-problems of size N/2



N/2

N/2

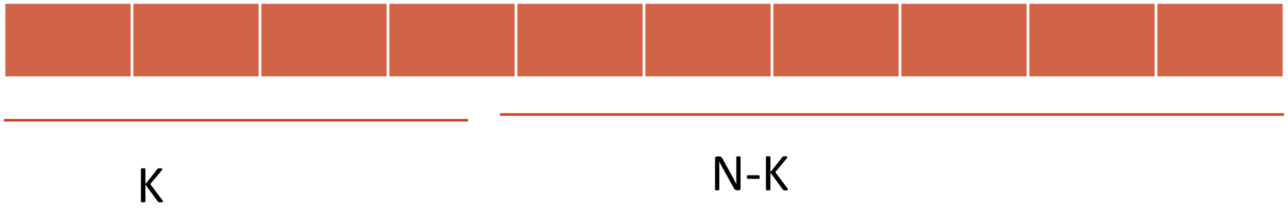
Where is this used? Provide examples.

For each example:

state the combination technique.

STRUCTURAL INDUCTION - EXAMPLE 1-C

- List / Array: Induction on size (say, N)
  - Consider sub-problems of size K and N-K, where K is variable.



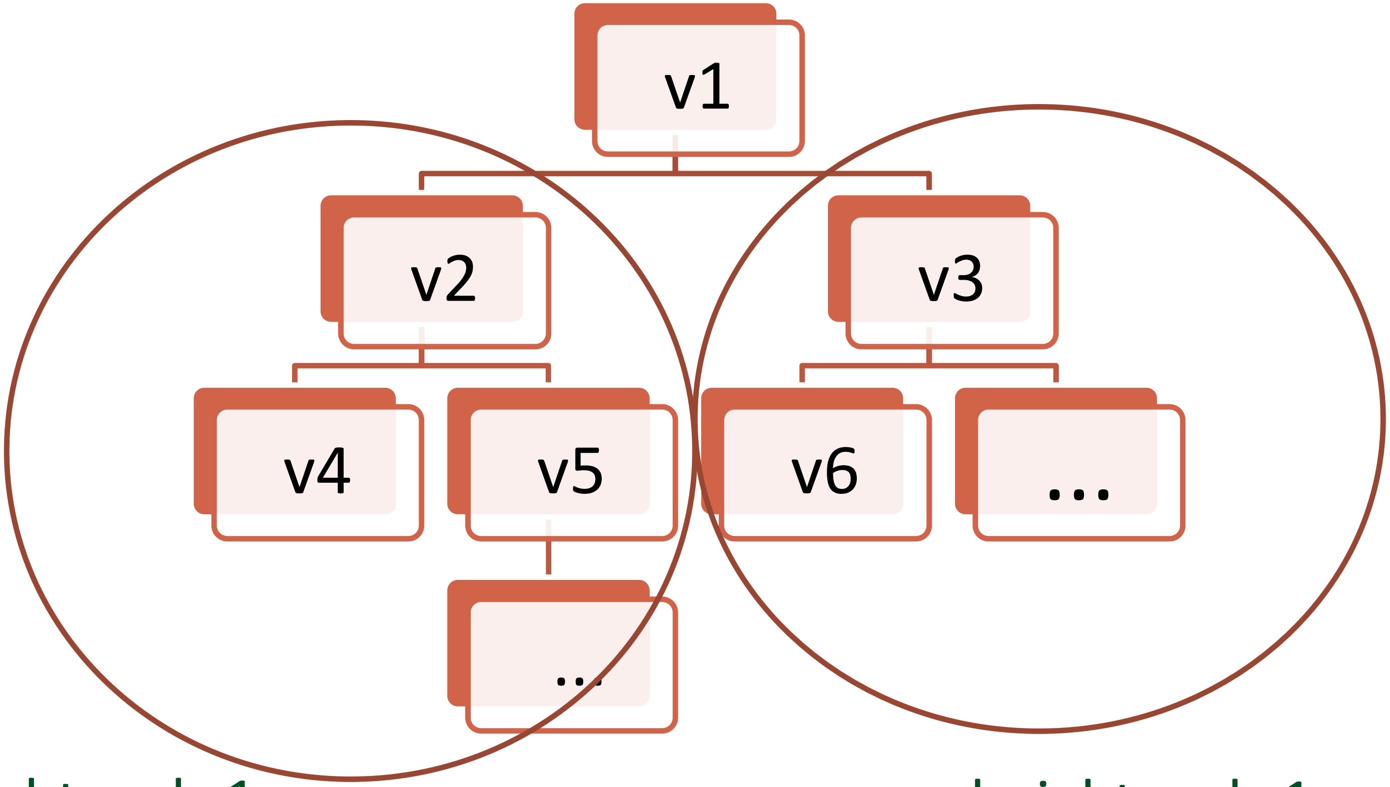
Where is this used?

Provide examples : provide values / ranges of K.

Contrast this with the scenario where K is a constant.

Structural Induction - Example 2

Binary Trees: **Induction on height (h)**



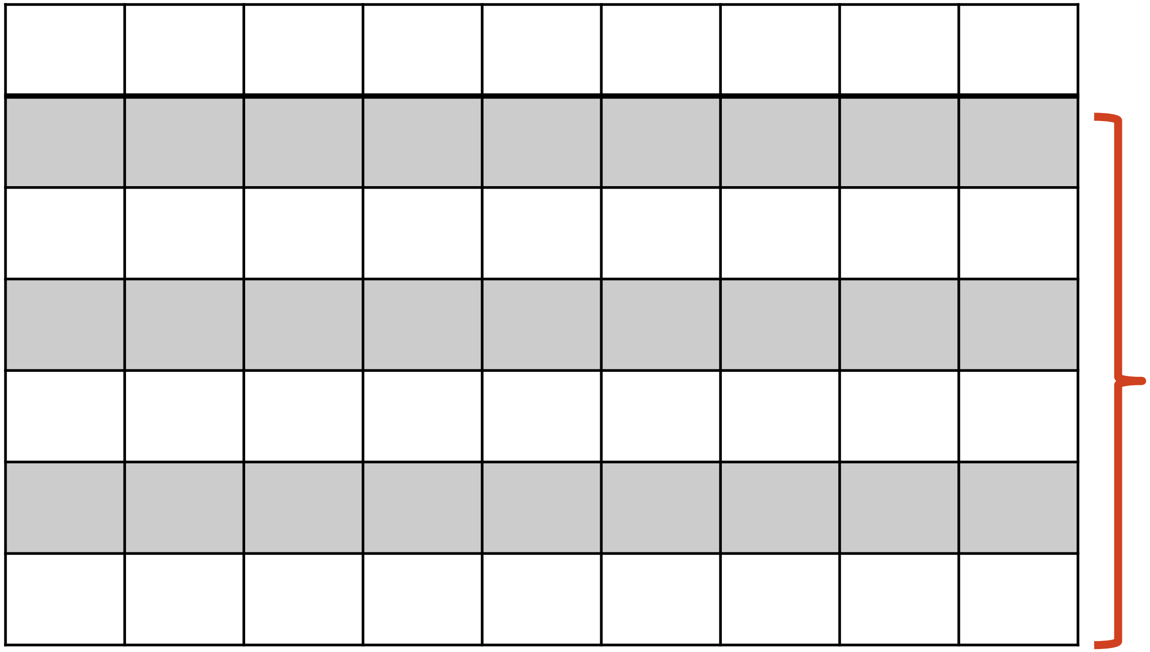
height <= h-1

height <= h-1

Structural Induction - Example 3

Matrix : Induction on size (say M*N)

Consider sub-problems of size 1*N and (M-1)*N



Row 1

M-1 Rows

What are other possible decompositions of a matrix?

DECOMPOSITION OF MATRICES

- Row-major vs. Column-major access
- Accessing rows / columns vs. accessing sub-matrices
  - Example problems where the latter is natural

Choice of Decomposition Of Matrices

- How do you choose between row-major vs. column major decompositions?
  - i.e. what does it depend on?
- The choice between row-major access vs. column-major access of a matrix is decided on the basis of the actual representation (storage)
  - i.e. whether the matrix is stored row-after-row or column-after-column in memory
    - Note that the memory is linear or one-dimensional.
- The cost of accessing data and the order of storage must align:
  - Locality of Reference(s)