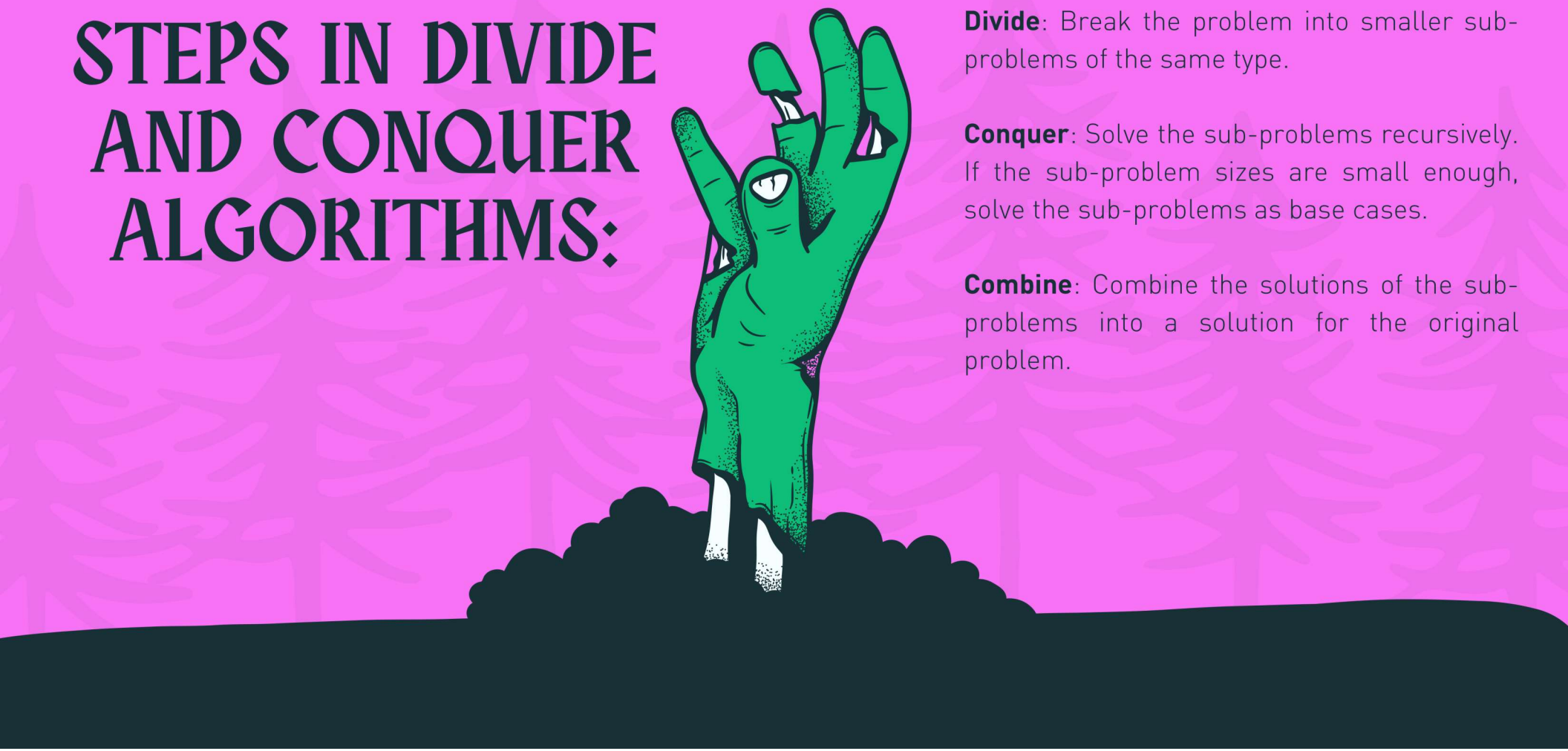

Divide and Conquer Algorthims
By KariAnn Harjo

STEPS IN DIVIDE AND CONQUER ALGORITHMS:



Divide: Break the problem into smaller sub-problems of the same type.

Conquer: Solve the sub-problems recursively. If the sub-problem sizes are small enough, solve the sub-problems as base cases.

Combine: Combine the solutions of the sub-problems into a solution for the original problem.

Characteristics

- These algorithms generally have a logarithmic or linearithmic complexity, making them very efficient for large datasets.
- They are recursive in nature.
- They are particularly useful when the problem can naturally be divided into independent sub-problems.

Example: Merge Sort

DIVIDE

Divide the unsorted list into n sub-lists, each containing one element (a list of one element is considered sorted).

CONQUER

Recursively divide the sub-lists in the previous step until there are multiple sorted sub-lists. Then, merge those sorted lists to produce new sorted lists.

COMBINE

Continue the process of merging until you get a single sorted list.

Big O Analysis

- Time Complexity: O(n log n), as the list is divided in half each time (log n divisions), and each division operation takes linear time (n).
- Space Complexity: O(n), as an additional space equivalent to the size of the array is required for the merging process.


Thank you!