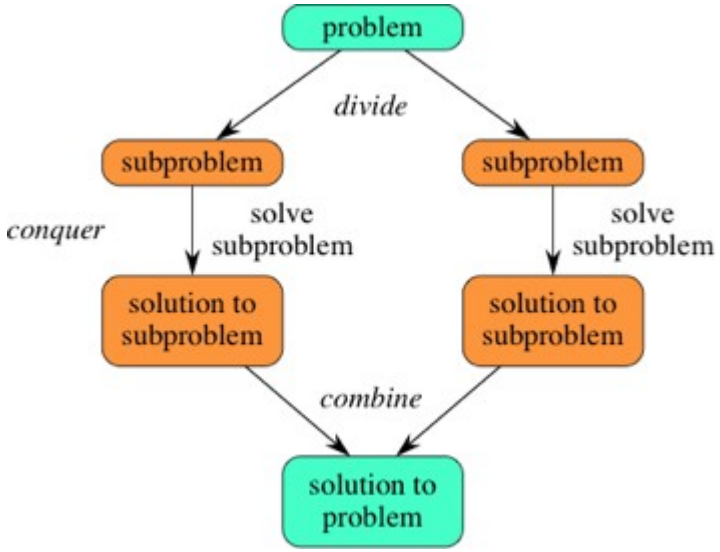
Divide & Conquer Approach

The Divide-and-Conquer Paradigm

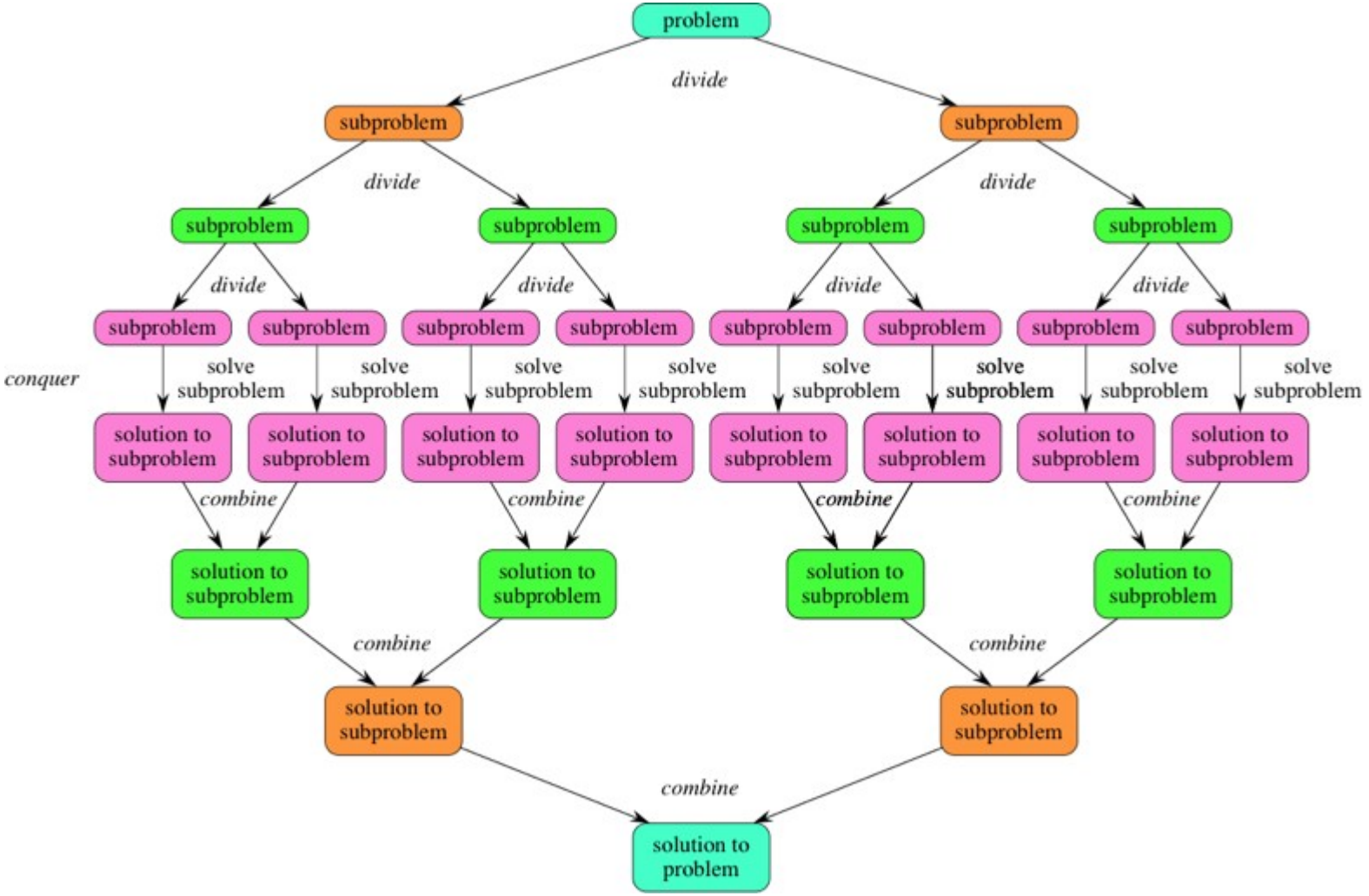
1. Divide the input into smaller subproblems.
2. Conquer the subproblems recursively.
3. Combine the solutions for the subproblems into a solution for the original problem.

```
DANDC(P) {
   if SMALL(P) then return S(p);
   else {
      divide P into smaller instances p1, p2,…, pk, k >= 1;
      apply DANDC to each of these sub problems;
      return (COMBINE(DANDC(p1), DANDC(p2),…, DANDC(pk));
   }
}
```

Divide & Conquer Approach



Divide & Conquer Approach



Divide & Conquer Approach (Master Method)

Standard Recurrence Format

Base case: $T(n)$ is at most a constant for all sufficiently small n.

General case: for larger values of n,

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O(n^d).$$

Parameters:

- $a =$ number of recursive calls
- $b =$ input size shrinkage factor
- $d =$ exponent in running time of the "combine step"

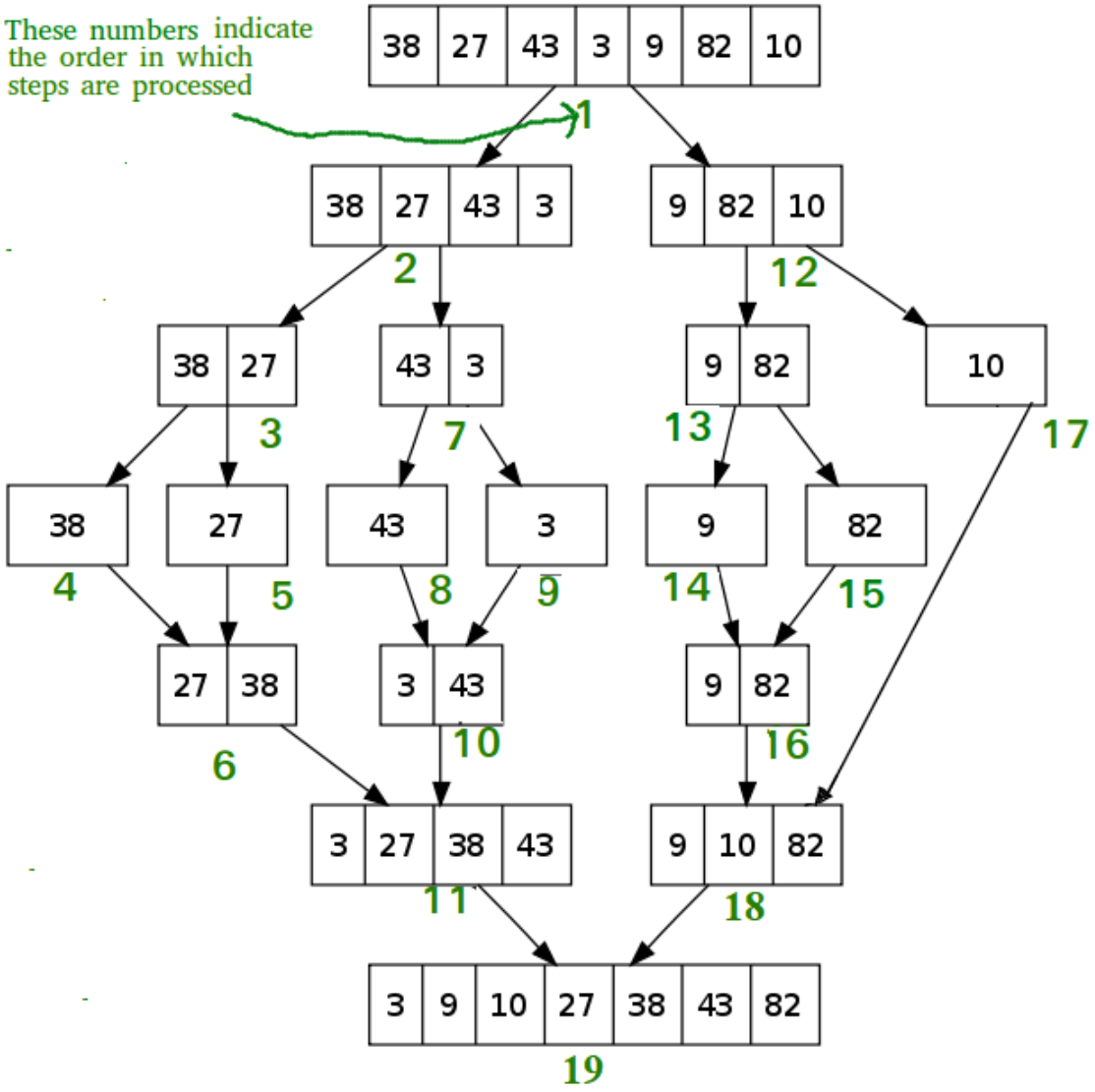
$$T(n) = \begin{cases} O(n^d \log n) & if\ a = b^d \\ O(n^d) & if\ a < b^d \\ O(n^{\log_b a}) & if\ a > b^d \end{cases}$$

Divide & Conquer Approach

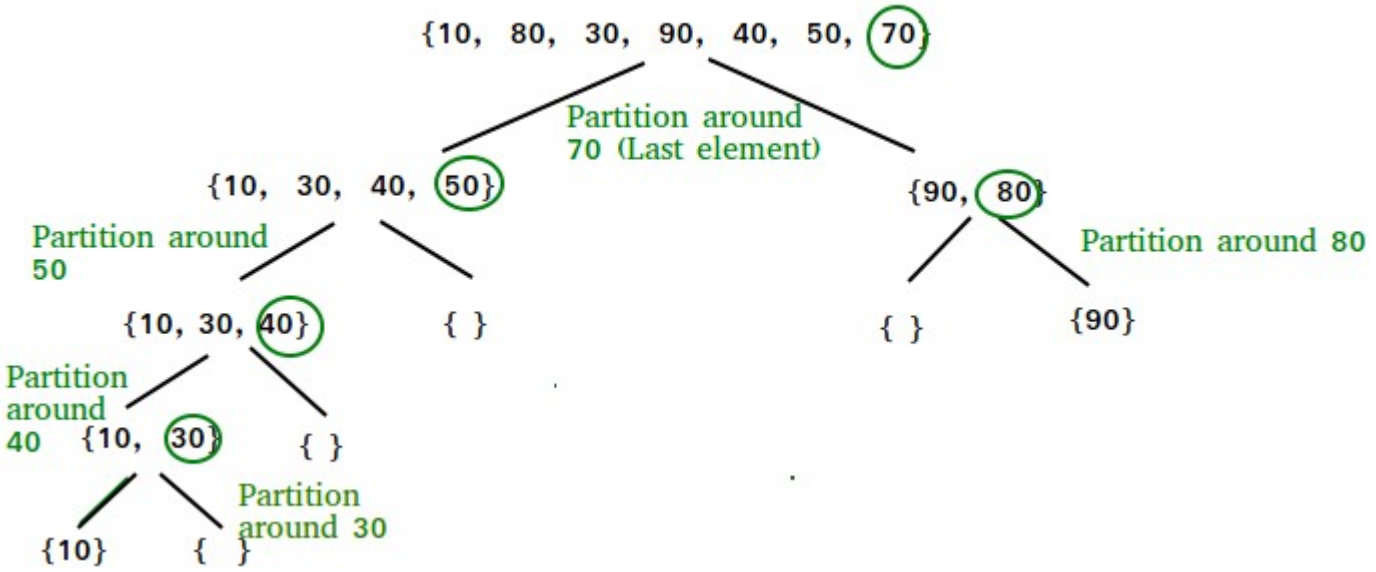
Sample applications

1. Mergesort
2. Quicksort
3. Closest pair of points
4. Strassen’s matrix multiplication
5. ...

Merge Sort


These numbers indicate the order in which steps are processed
38 27 43 3 9 82 10
1
38 27 43 3
9 82 10
2
12
38 27
43 3
9 82
10
3
7
13
17
38
27
43
3
9
82
4
5
8
9
14
15
27 38
3 43
9 82
10
16
6
3 27 38 43
9 10 82
11
18
3 9 10 27 38 43 82
19

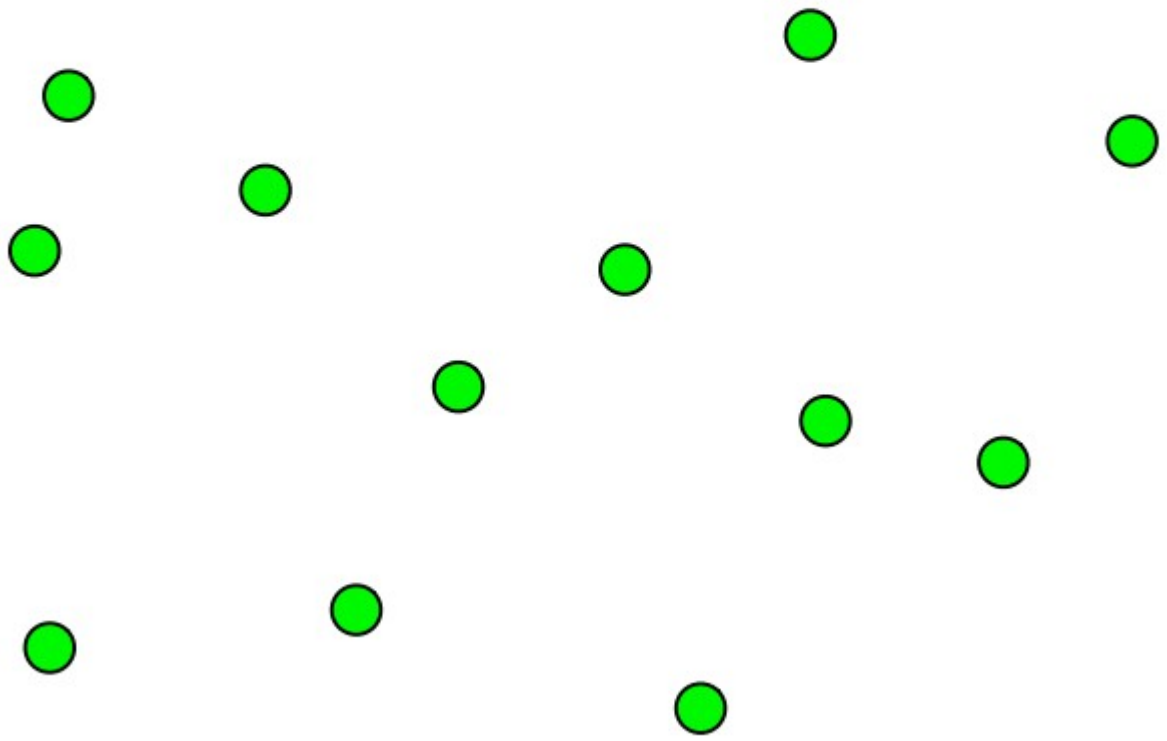
Quick Sort


{10, 80, 30, 90, 40, 50, 70}
Partition around 70 (Last element)
{10, 30, 40, 50}
{90, 80}
Partition around 50
Partition around 80
{10, 30, 40}
{ }
{ }
{90}
Partition around 40
{10, 30}
{ }
Partition around 30
{10}
{ }

Closest Pair of Points

Problem

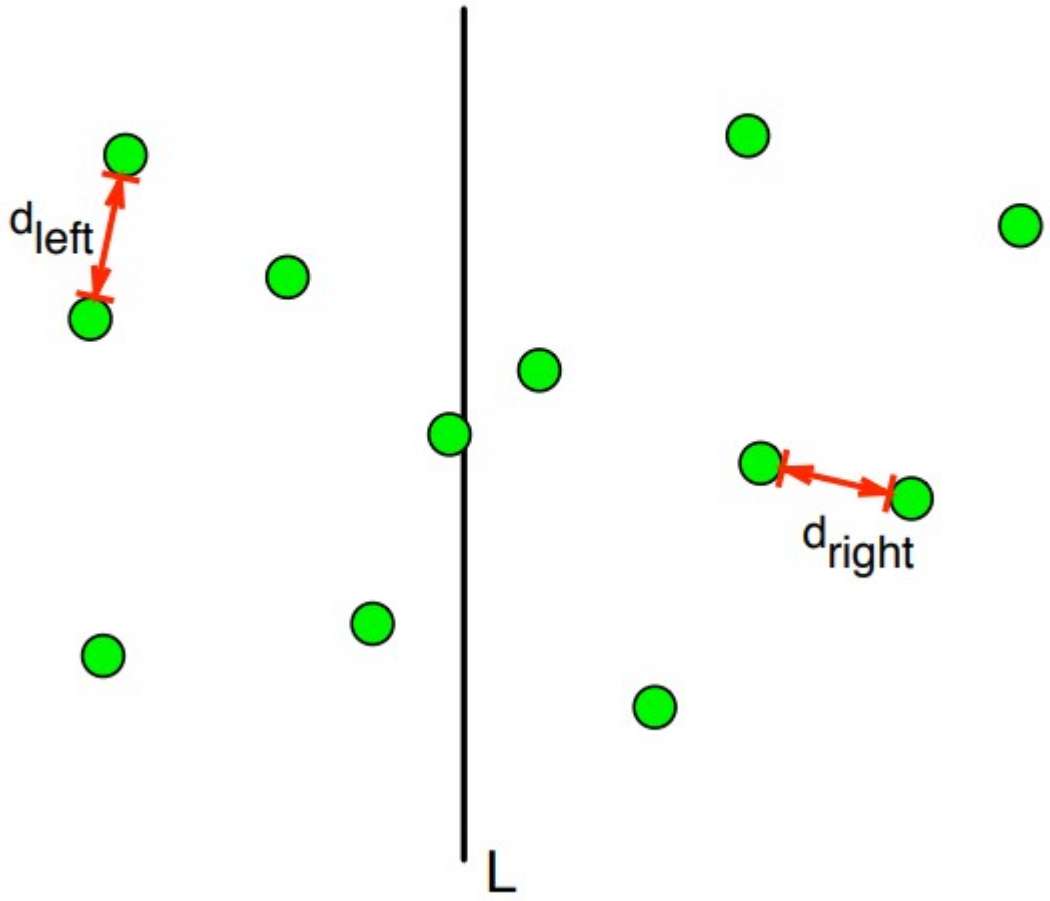
Given a set of points $\{p_1, \ldots, p_n\}$ find the pair of points $\{p_i, p_j\}$ that are closest together.



Closest Pair of Points

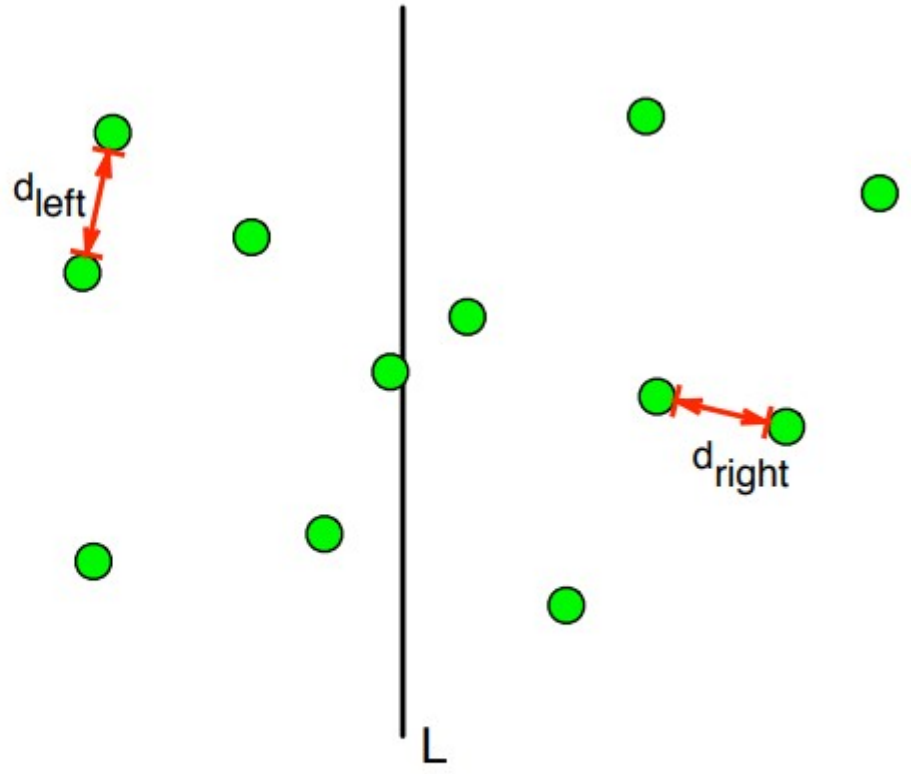
Split the points with line L so that half the points are on each side.

Recursively find the pair of points closest in each half.



Closest Pair of Points

Let $d = \min\{d_{\text{left}}, d_{\text{right}}\}$.



- d would be the answer, except maybe L split a close pair!

Closest Pair of Points

If there is a pair $\{p_i, p_j\}$ with $\mathbf{dist}(p_i, p_j) < d$ that is split by the line, then both p_i and p_j must be within distance d of L.