

Brute Force

↳ A straightforward approach to solving a problem, usually directly based on the problem statement and definitions of the concepts involved.

0 Selection Sort

0 Bubble Sort

how many copies in a swap? ③

A first application of a brute force algorithm often results in one that can be improved with a modest amount of effort.

o Sequential Search

o Brute Force String Matching:

(Brute Force)

↳ Finite set of points in a plane

• Closest Pair Problem

→ Euclidean distance:

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

```
ALGORITHM BruteForceClosestPair(P)
  //Finds distance between two closest points in the plane by brute force
  //Input: A list P of n ( $n \geq 2$ ) points  $p_1(x_1, y_1), \dots, p_n(x_n, y_n)$ 
  //Output: The distance between the closest pair of points
   $d \leftarrow \infty$ 
  for  $i \leftarrow 1$  to  $n - 1$  do
    for  $j \leftarrow i + 1$  to  $n$  do
       $d \leftarrow \min(d, \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)})$  //sqrt is square root
  return  $d$ 
```

The basic operation here is square of a number. Since we can just compare the values to themselves. (no need for square root!)

→ Hamming distance:

- Convex Hull Problem

- Convex:

- Convex Hull:

- Convex-hull - problem:

- Extreme point:

- Linear programming:

- Traveling Salesman Problem

- Find the best Hamiltonian Circuit for the graph.

- knapSack Problem

- highest value for max weight

- Assignment Problem

* Depth First Search