## Brute Force

→ Straight fwd approach to solving a problem. usually directly based on problem statement & definitions of concepts involved.

- → "force"
- → "just do it"

## closest-pair & convex-hull problems

↳ Straight fwd

→ pblms deals with finite set of points

→ areas:- computational, geometry & OR

## closest-pair problem

→ finding 2-closest points in set of n points.

points are (x, y)

cartesian co-ordinates

↳ distance ~~pont~~ btw 2 points

$P_i = (x_i, y_i)$ & $P_j = (x_j, y_j)$

Std Euclidean distance

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

gorithm_Brute force closest points (P)
// finds 2 closest points in plane by brute force
// I/P : list P of n ($n \geq 2$) points $P_1 = (x_1, y_1)$ ... $P_n = (x_n, y_n)$
// O/P : indicate index1 & index2 of closest pair

```
dmin ← ∞
for i ← 1 to n-1 do
    for j ← i+1 to n do
        d ← sqrt((x_i - x_j)^2 + (y_i - y_j)^2)
        if d < dmin
            dmin ← d; index1 ← i; index2 ← j
return index1, index2
```

$$C(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 2 = 2 \sum_{i=1}^{n-1} (n-i)$$

$$= 2[(n-1) + (n-2) + \cdots + 1] = (n-1)n \in \Theta(n^2)$$