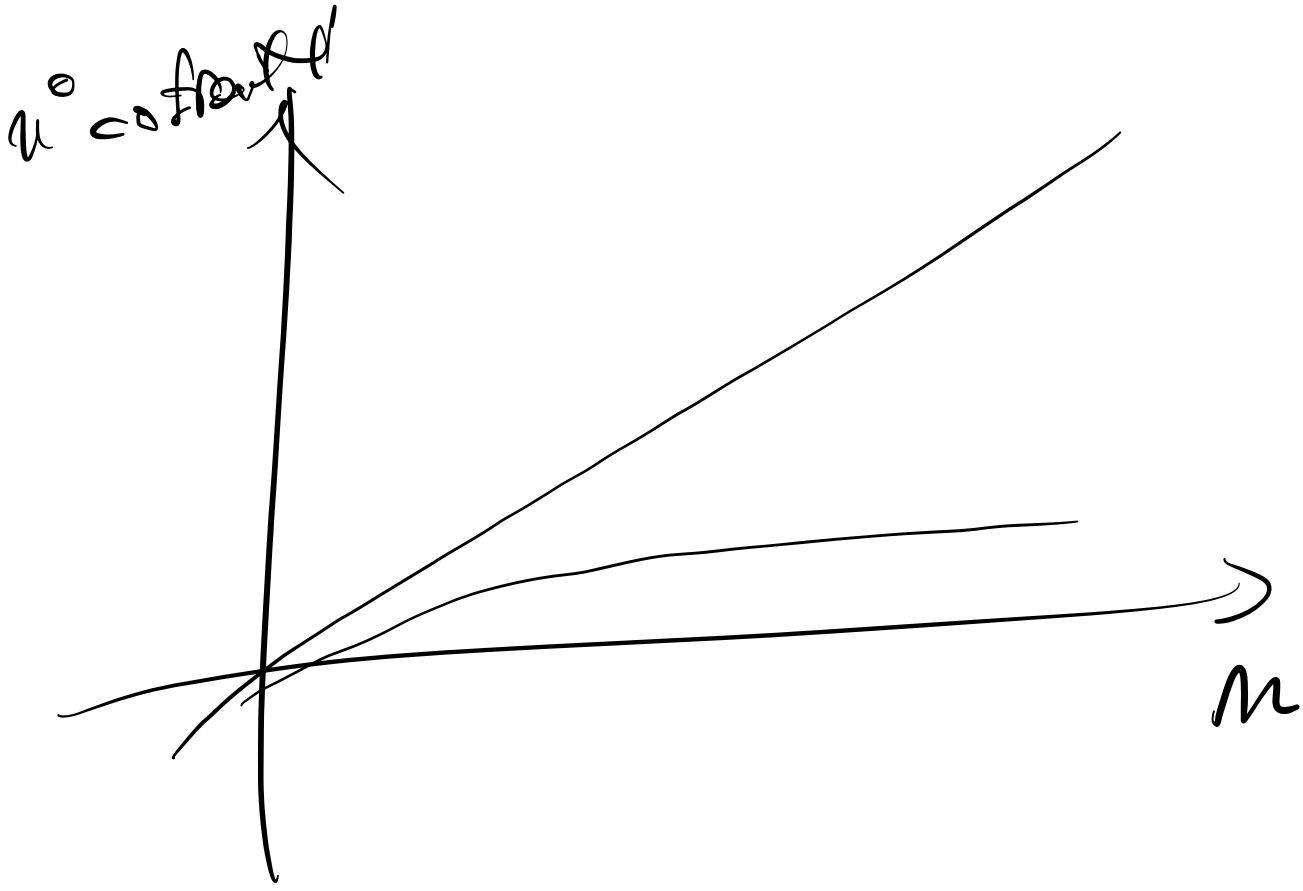
SORTING & SEARCHING

```
var s []string
```

Giovanni	Filippo	Amleto	Susanna

Gigi ∈ s

```
for _, x := range s {
    if x == "Gigi" {
        return true
    }
}
return false
```



RICERCA DICOTOMICA IN UNA SLICE ORDINATA

```
func BinarySearch(s []string, x string) bool {

    left := 0
    right := len(s) - 1
    for left <= right {
        m := (left+right)/2
```

```
mid := (left + right) / 2
if s[mid] == x {
    return true
}
if x < s[mid] {
    right = mid - 1
} else {
    left = mid + 1
}
}
return false
}
```

$$n \quad \frac{n}{2} \quad \frac{n}{4} \quad \cdots \quad \frac{n}{2^k} \leq 1$$

$$\frac{n}{2^k} = 1$$

$$2^k$$

$n = 2^k$

$\log_2 n = k$

ORDINAMENTO

S [] string

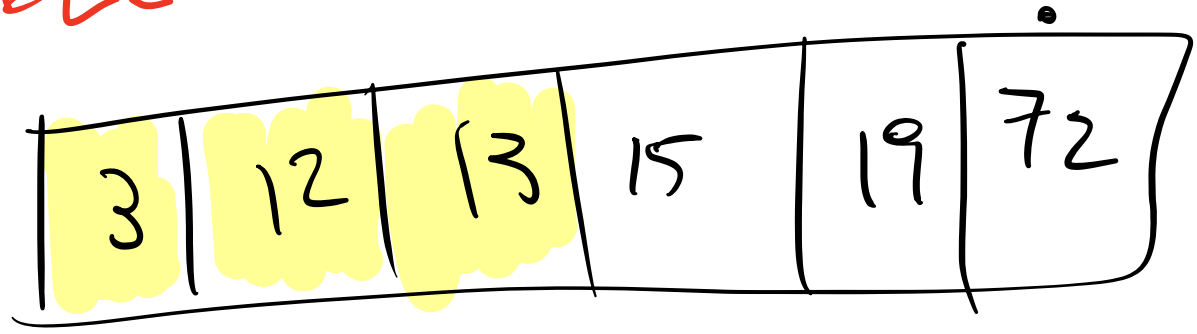
Ibbù on <=

- in-place ④
  - non in-place
- stabile
  - non stabile
- confronti & scambi ③
  - radix-based

naif $O(n^2)$

A B — sofisticati $O(n \lg n)$

SELECTION SORT



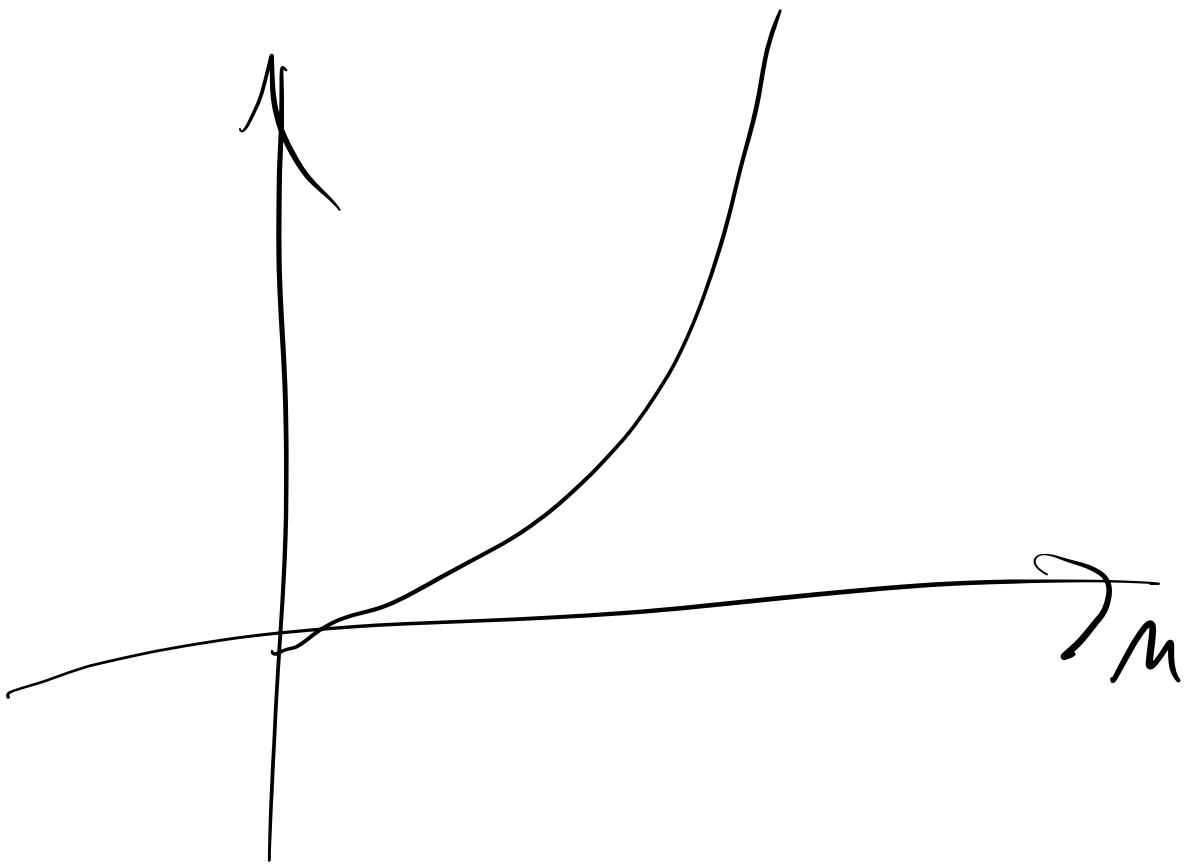

```
func selectionSort(s []string) {
    n := len(s)
    for i := 0; i < n; i++ {
        // Devo cercare cosa
        // mettere in posizione i
        minIndex = i
        for j = i+1; j < n; j++ {
            if s[j] < s[minIndex] {
                minIndex = j
            }
        }
    }
```

s[i], s[minIndex] = s[minIndex], s[i]

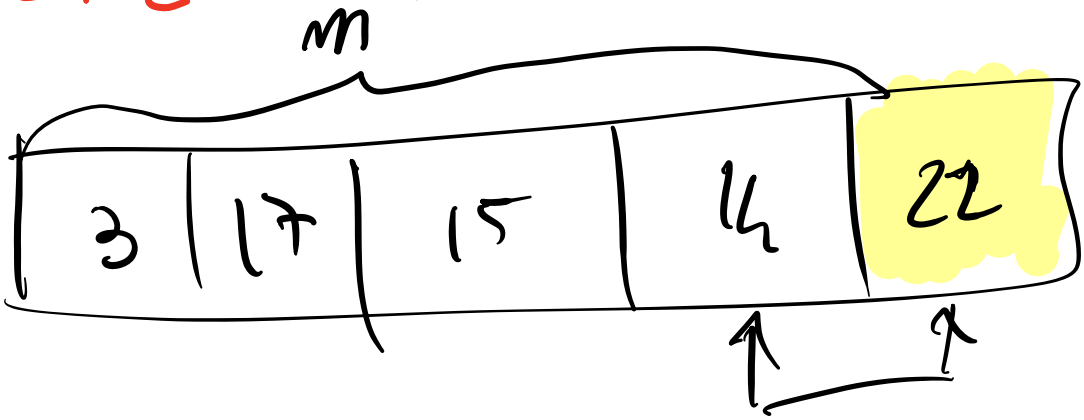
i	
0	n-1
1	n-2
2	n-3
⋮	
n-1	0

$$0+1+\dots+(n-1)=\frac{n(n-1)}{2}$$

$$=\frac{1}{2}n^2-\frac{1}{2}n$$



BUBBLESORT




```
fun BubbleSort(s []string) {
    n = len(s)
    for m = n; m > 0; m-- {
        for i = 0; i < m-1; i++ {
            if s[i] > s[i+1] {
                s[i], s[i+1] = s[i+1], s[i]
            }
        }
    }
}
```

m	
n	n-2
	n-3

$0+1+\dots+(n-2) = \frac{(n-2)(n-1)}{2} =$

$n-1$
$n-2$
$\vdots$
1

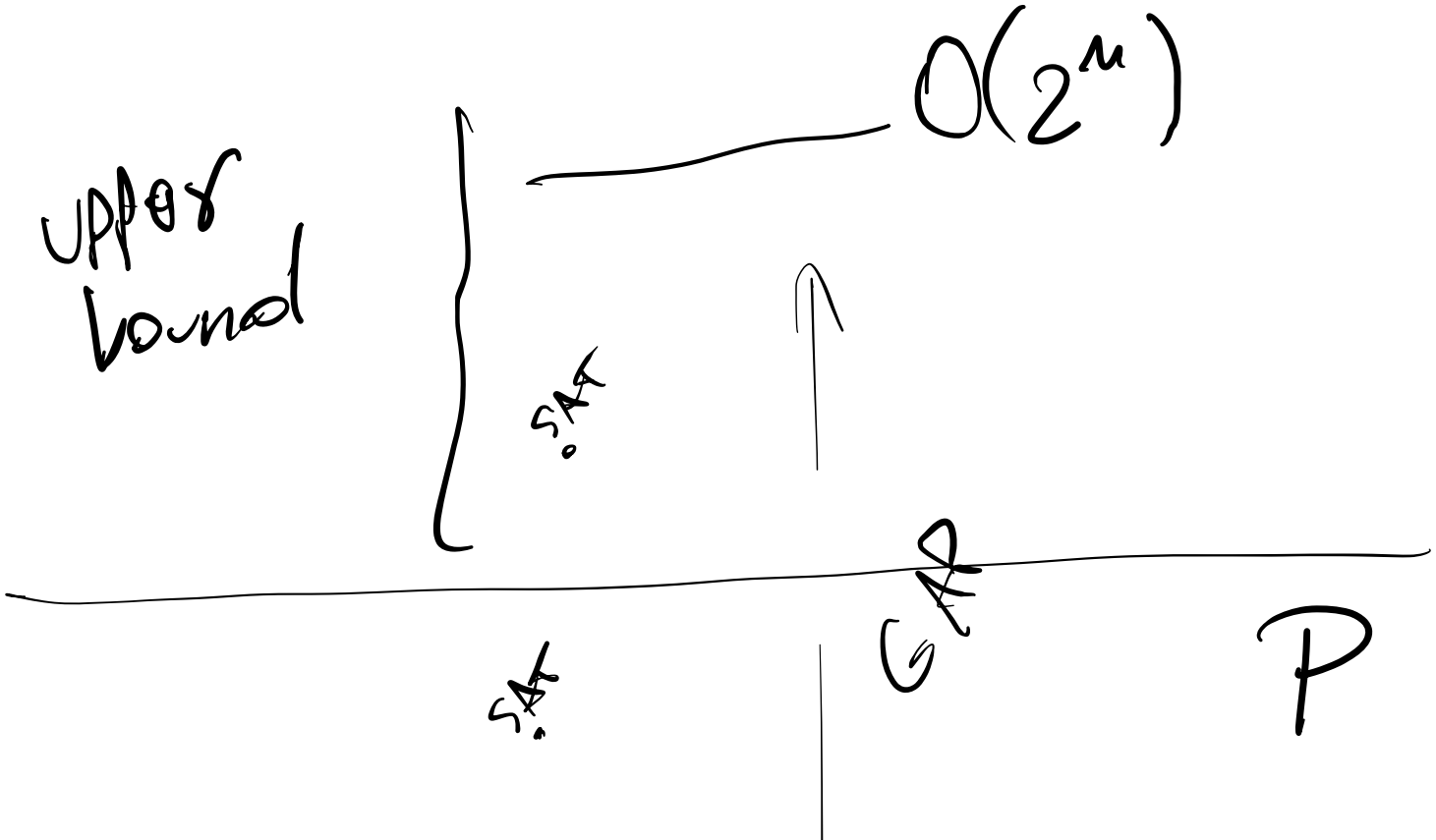
$n-4$
$\vdots$
0

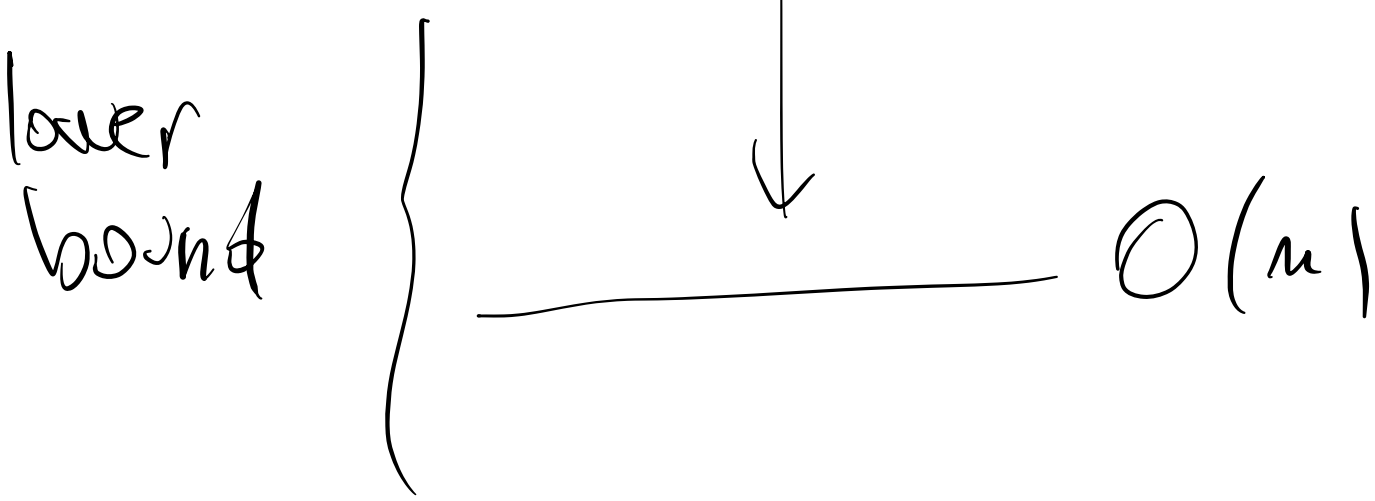
$= \frac{1}{2}n^2 - \frac{3}{2}n + \frac{3}{2}$

$P \stackrel{?}{=} NP$

Problema SAT

$(x \wedge y \wedge \neg z) \vee (x \wedge \neg y \wedge w) \ldots$




lower
bound
O(n)