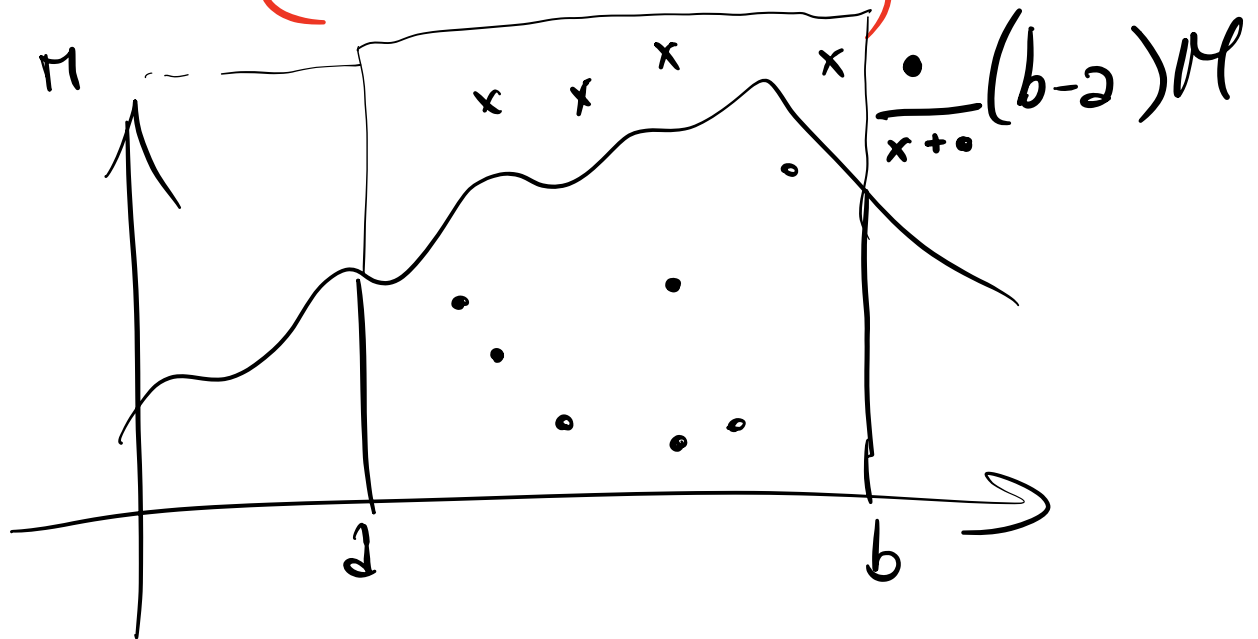


ANALISI NUMERICA

INTEGRAZIONE MONTECARLO (HIT-OR-MISS)



ORDINAMENTO DI SLICE

TACP

17	1	35	12	44
----	---	----	----	----

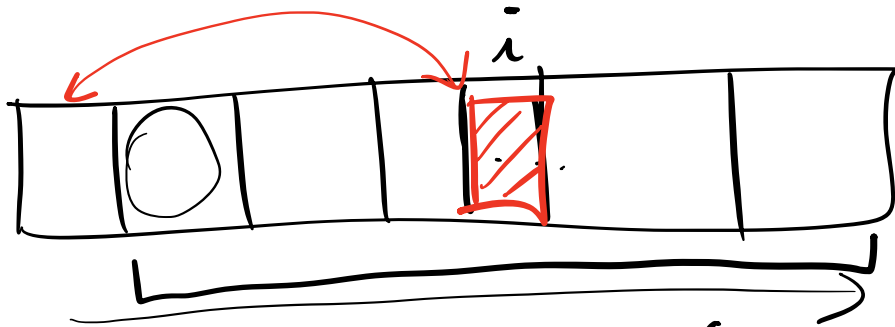
↓ "sol part"

1	12	17	35	44
---	----	----	----	----

$O(N^2)$

$$O(n \lg n)$$

SELECTION SORT



$$n + (n-1) + (n-2) + \dots + 1$$

$$= \frac{n(n+1)}{2} = \frac{1}{2} n^2 + \frac{1}{2} n$$

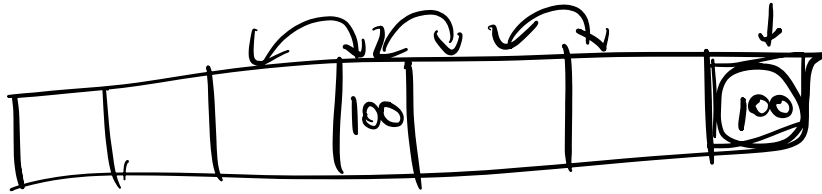
func selectionSort (x int[]) {
 n := len(x)
 for i = 0; i < n; i++ {
 // slice x[i:] is disordered
 // where x[i] is ordered
 min = x[i]
 indexMin = i
 for j = i+1; j < n; j++ {
 if x[j] < min {
 min = x[j]
 indexMin = j
 }
 }
 // swap x[i] with x[indexMin]
 }
}

$x[i], x[\text{index}(i)] = x[\text{index}(i)], x[i]$

}

}

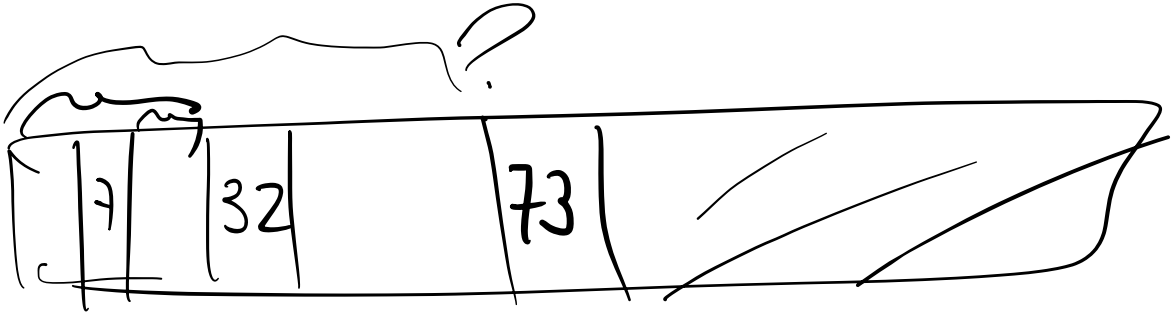
BUBBLE SORT



```
func bubbleSort(x int[]) {  
    n := len(x)  
    for i := n; i > 0; i-- {  
        x[i:] è ordinato  
        e contiene gli elementi  
        più piccoli  
        swap := false  
        for j := 0; j < i; j++ {  
            if x[j] > x[j+1] {  
                x[j], x[j+1] =  
                    x[j+1], x[j]  
                swap = true  
            }  
        }  
        if !swap { return }  
    }  
}
```

}

RICERCA DICOTOMICA



15

$$n \quad \frac{n}{2} \quad \left(\frac{n}{4} \right)$$

N° confronti è il minimo
* t.c.

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

$$n = 2^k$$

$$\lg_2 n = \lg_2 2^k$$

$$k = \lg_2 n$$