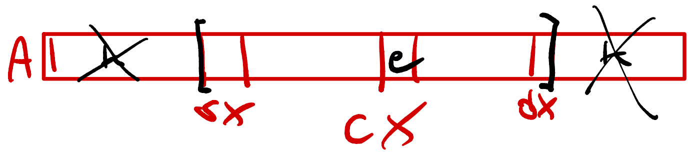
RICERCA BINARIA

IDEA



chiave k

$$cx = \frac{sx+dx}{2}$$

1) $e = k$: ok
2) $e > k$: escludiamo $A[cx..dx]$
3) $e < k$: escludiamo $A[sx..cx]$

In realtà vogliamo trovare la posizione più a sinistra che contiene k

A = [1 2 3 4 4 4 4 4 5 6 6 6 7 8]

0 1 2 3 4 5 6 7 8 9 10 11 12 13

sx cx dx

K = 4

DIVIDE ET IMPERA

```
RicercaBinariaRicorsiva( a, k, sinistra, destra ):
                          ⟨pre: a ordinato e 0 ≤ sinistra ≤ destra ≤ n - 1⟩
  IF (sinistra == destra) {
    IF (k == a[sinistra]) {
      RETURN sinistra;
    } ELSE {
      RETURN -1;
    }
  }
  centro = (sinistra+destra)/2;
  IF (k <= a[centro]) {
    RETURN RicercaBinariaRicorsiva( a, k, sinistra, centro );
  } ELSE {
    RETURN RicercaBinariaRicorsiva( a, k, centro+1, destra );
  }
```

k non appare

caso base $n_0 = 1$

DIVIDE

IMPERA

OR (esclusivo)

IMPERA

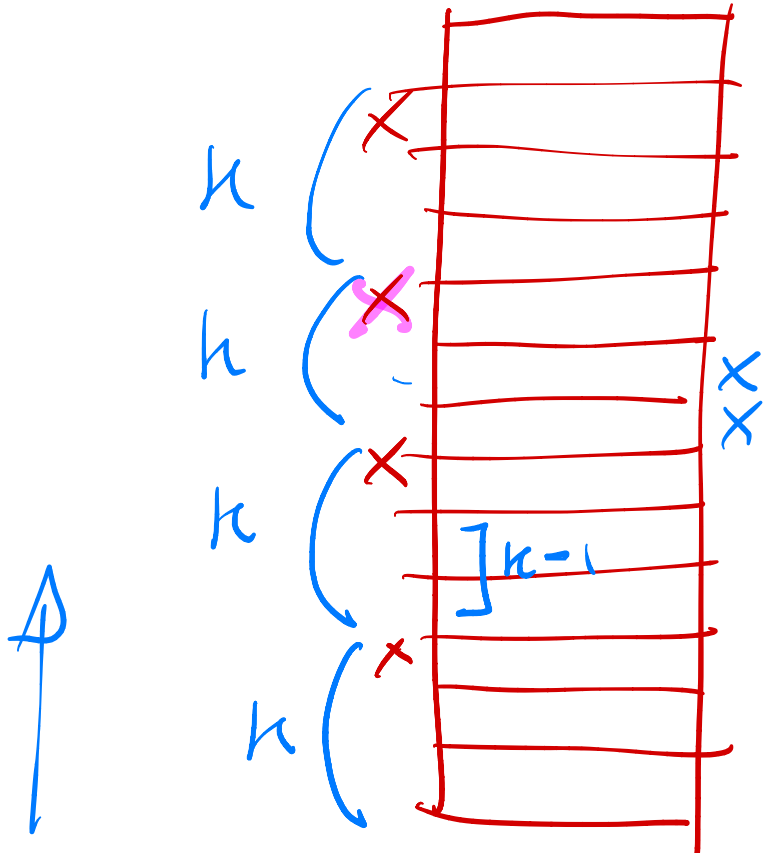
$O(\lg n)$ tempo

$n \to n/2 \to n/4 \to \cdots \to n/2^{\ell} = 1$

$\Updownarrow$

$\ell = \lg n$

n piani



$k=3$

$$\boxed{\frac{n}{k} + k - 1}$$

$$\frac{n}{k} = k - 1 \quad \text{migliore}$$

$$\sim \; k \sim \sqrt{n}$$

BASSO

3 uova

$$\frac{n}{k} + \underbrace{\sqrt{k-1}}_{k-1} \qquad \min$$

1 uovo   2 uova

$$k \sim n^{1/3}$$

4 uova

$$\frac{n}{k} + (k-1)^{1/3} \qquad \min$$

$$k \sim n^{1/4}$$

$$n^{1/u} = 2 \iff 2^{\log_2 n / u} = 2 \iff u = \log_2 n$$

```
MergeSort( a, sinistra, destra ):
                                        ⟨pre: 0 ≤ sinistra ≤ destra ≤ n - 1⟩
O(1)   IF (sinistra < destra) {
O(1)     centro = (sinistra+destra)/2;
T(n/2)   MergeSort( a, sinistra, centro );
T(n/2)   MergeSort( a, centro+1, destra );
O(n)     Fusione( a, sinistra, centro, destra );
       }
```

$n = dx - sx + 1$ = numero di elementi

$T(n)$ = costo di Mergesort

?

$$T(n) \leq \begin{cases} \text{costante} & \text{per } n \leq n_0 \\ 2T(n/2) + cn & \text{per } n > n_0 \end{cases}$$

c = costante > 0

Relazione di ricorrenza

```
RicercaBinariaRicorsiva( a, k, sinistra, destra ):
                              ⟨pre: a ordinato e 0 ≤ sinistra ≤ destra ≤ n - 1⟩
       IF (sinistra == destra) {
         IF (k == a[sinistra]) {
           RETURN sinistra;
O(1)     } ELSE {
           RETURN -1;
         }
       }
O(1)   centro = (sinistra+destra)/2;
O(1)   IF (k <= a[centro]) {
T(n/2)   RETURN RicercaBinariaRicorsiva( a, k, sinistra, centro );
       } ELSE {
T(n/2)   RETURN RicercaBinariaRicorsiva( a, k, centro+1, destra );
       }
```

$$T(n) \leq T(n/2) + c$$

$$T(n) \leq 2T(n/2) + c$$

```
MAXR(A, sx, dx):
        n = dx - sx + 1
        if n ≤ 0:
O(1)      return -∞
        if n = 1: // sx = dx
          return A[sx]
        // n ≥ 2
O(1)    cx = (sx + dx)/2
T(n/2)  (1) m1 = MAXR(A, sx, cx)
T(n/2)  (2) m2 = MAXR(A, cx+1, dx)
O(1)    return max(m1, m2)
```

Teorema 3.1 *Sia* $f(n)$ *una funzione non decrescente e siano* α, β, n_0, c_0 *e* c *delle costanti tali che* $\alpha \geq 1, \beta > 1$ *e* $n_0, c_0, c > 0$*, per la relazione di ricorrenza*

$$T(n) \leq \begin{cases} c_0 & \textit{se } n \leq n_0 \\ \alpha T(n/\beta) + \boxed{cf(n)} & \textit{altrimenti} \end{cases} \quad (3.2)$$

(dove n/β *va interpretato come* $\lfloor n/\beta \rfloor$ *o* $\lceil n/\beta \rceil$*). Se esistono due costanti positive* γ e n'_0 tali che $\boxed{\alpha f(n/\beta) \leq \gamma f(n)}$ *per ogni* $n \geq n'_0$*, allora la relazione di ricorrenza ha i seguenti limiti superiori:*

1. $T(n) = O(f(n))$ *se* $\gamma < 1$;
2. $T(n) = O(f(n) \log_\beta n)$ *se* $\gamma = 1$;
3. $T(n) = O(n^{\log_\beta \alpha})$ *se* $\gamma > 1$ e $\alpha > 1$.

MS: $\alpha = 2, \beta = 2, f(n) = n$
RB: $\alpha = 1, \beta = 2, f(n) = 1$
MAX: $\alpha = 2, \beta = 2, f(n) = 1$

1) l'algoritmo invoca la ricorsione su n/β elementi
2) ci sono α chiamate ricorsive
3) Ogni chiamata richiede $\leq c\, \boxed{f(n)}$ tempo (escluse le chiamate ricorsive)

Teorema 3.1 *Sia* $f(n)$ *una funzione non decrescente e siano* α, β, n_0, c_0 *e* c *delle costanti tali che* $\alpha \geq 1$, $\beta > 1$ *e* $n_0, c_0, c > 0$, *per la relazione di ricorrenza*

$$T(n) \leq \begin{cases} c_0 & \text{se } n \leq n_0 \\ \alpha T(n/\beta) + cf(n) & \text{altrimenti} \end{cases} \tag{3.2}$$

(dove n/β *va interpretato come* $\lfloor n/\beta \rfloor$ *o* $\lceil n/\beta \rceil$*). Se esistono due costanti positive* γ e n'_0 tali che $\alpha f(n/\beta) \leq \gamma f(n)$ *per ogni* $n \geq n'_0$, *allora la relazione di ricorrenza ha i seguenti limiti superiori*:

1. $T(n) = O(f(n))$ *se* $\gamma < 1$;
2. $T(n) = O(f(n)\log_\beta n)$ *se* $\gamma = 1$;
3. $T(n) = O(n^{\log_\beta \alpha})$ *se* $\gamma > 1$ e $\alpha > 1$.

MS: $\alpha = 2, \beta = 2, f(n) = n$
RB: $\alpha = 1, \beta = 2, f(n) = 1$
MAX: $\alpha = 2, \beta = 2, f(n) = 1$

MS: $\alpha f(n/\beta) \leq \gamma f(n)$

$2 f(n/2) \leq \gamma f(n)$

$2 \cdot n/2 \leq \gamma n \Rightarrow \gamma = 1 \overset{2.}{\Rightarrow} T(n) = O(f(n) \log_\beta n) = O(n \log n)$

RB: $\alpha f(n/\beta) \leq \gamma f(n)$

$1 \cdot 1 \leq \gamma \cdot 1 \Rightarrow \gamma = 1 \overset{2.}{\Rightarrow} T(n) = O(1 \cdot \log_2 n) = O(\log n)$

MAX: $\alpha f(n/\beta) \leq \gamma f(n)$

$2 \cdot 1 \leq \gamma \cdot 1 \Rightarrow \gamma = 2 \overset{3.}{\Rightarrow} T(n) = O(n^{\log_\beta \alpha}) = O(n^{\log_2 2}) = O(n)$