

TECNICO SUPERIORE WEB DEVELOPER FULL STACK

#4 - Algoritmi di ricerca

Definizione

Un algoritmo di ricerca è un algoritmo che, data una sequenza di elementi, restituisce la posizione di un elemento cercato, se presente, o un valore nullo (tipicamente -1) se non è presente.

In alternativa, può anche essere una funzione booleana che restituisce true o false a seconda che l'elemento sia o meno presente.

La ricerca “intuitiva”: sequenziale

Il modo più semplice di cercare un elemento in una sequenza è quello di scorrere ogni componente e verificare se corrisponda o meno a quello cercato.

Una volta raggiunta la fine della sequenza siamo sicuri che l'elemento non è presente.

Iterativamente...

```
int LinearSearch(int[] a, int x) {  
    for(int pos = 0; pos < a.Length; pos++) {  
        if(a[pos] == x)  
            return pos;  
    }  
    return -1;  
}
```

Ricorsivamente...

```
int LinearSearch(int[] a, int x, int pos) {  
    if (pos == a.Length)  
        return -1;  
    else if (a[pos] == x)  
        return pos;  
    else  
        return LinearSearch(a, x, pos+1);  
}
```

Però...

Questo approccio è valido in generale, nel senso che in mancanza di altre informazioni sugli elementi della collezione (in particolare sul loro ordinamento) va bene sempre.

Quanto è efficiente?

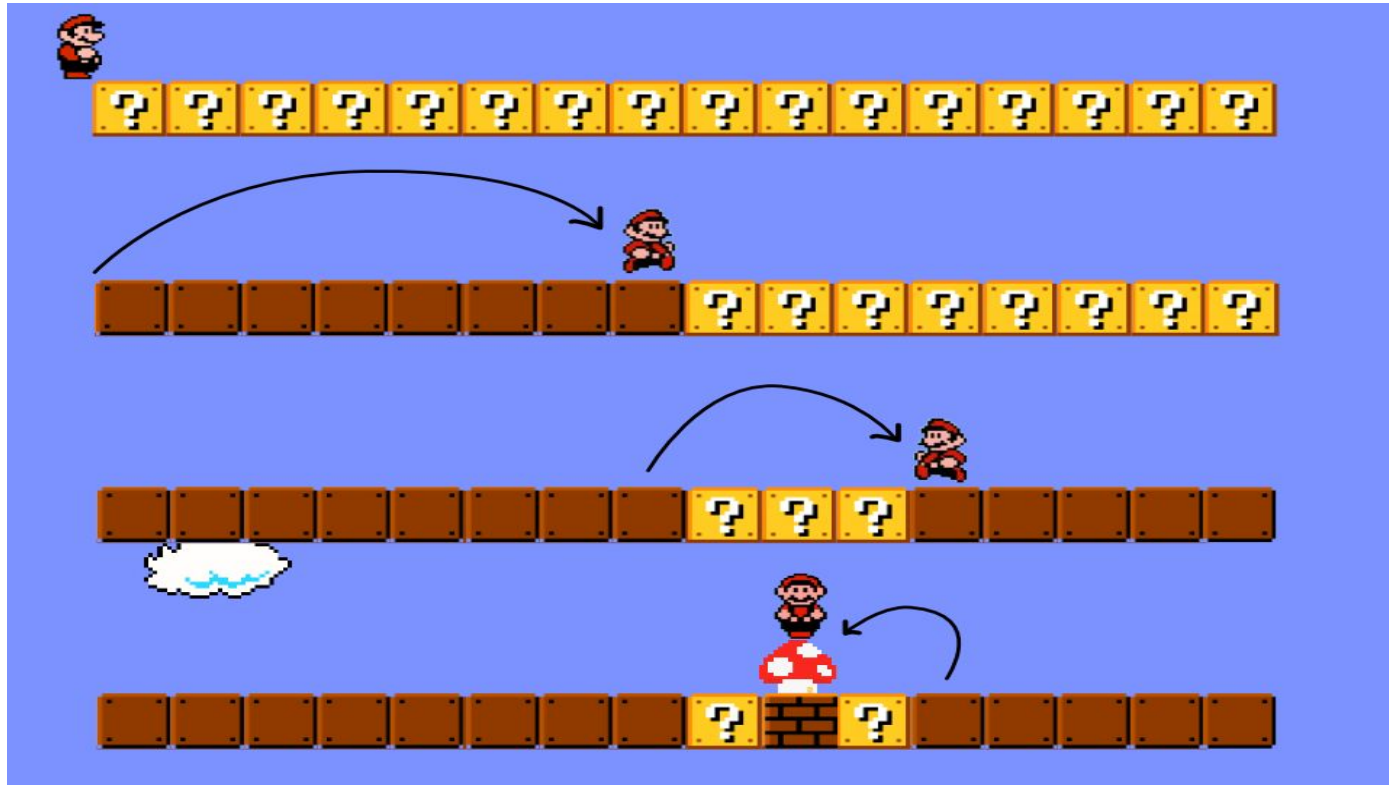
Senza entrare nei dettagli di “cos’è la complessità”, possiamo intuire che, siccome dobbiamo scorrere tutta la sequenza, più essa è lunga, più tempo impieghiamo.

Si può fare di meglio?

Dipende:

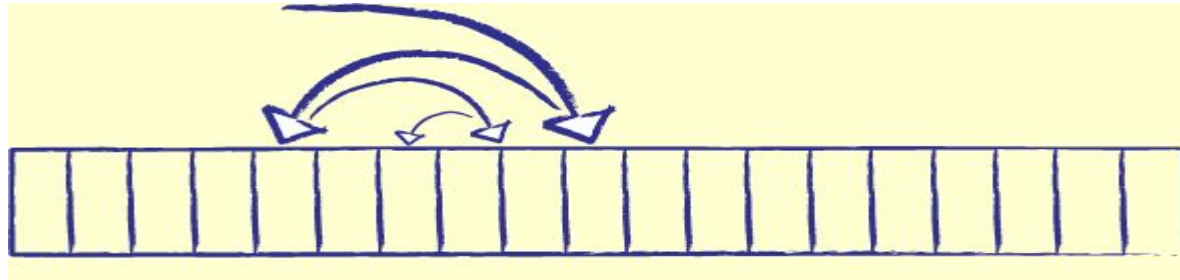
- in mancanza di altre informazioni, no;
- se invece sappiamo che la sequenza è ordinata, possiamo sfruttare questa informazione per migliorare l'algoritmo.

Ricerca binaria, o dicotomica



Funzionamento

1. Inizia considerando la sequenza per intero
2. Osserva l'elemento a metà
3. Se è l'elemento cercato, FINE
4. Se l'elemento cercato è maggiore, sarà nella metà superiore
5. Se l'elemento cercato è minore, sarà nella metà inferiore
6. Torna al punto 2



Esempio numerico

If searching for 23 in the 10-element array:

2	5	8	12	16	23	38	56	72	91
---	---	---	----	----	----	----	----	----	----

23 > 16,
take 2nd half

L					H				
2	5	8	12	16	23	38	56	72	91

23 < 56,
take 1st half

L					H				
2	5	8	12	16	23	38	56	72	91

Found 23,
Return 5

L					H				
2	5	8	12	16	23	38	56	72	91

Pseudo - codice

Iterativo:

1. Imposta due indici, inizialmente in cima e in fondo alla sequenza
2. Finché non si “scambiano di posto”
3. Controlla l'elemento in mezzo
4. Se il valore cercato è minore, sposta indietro l'indice di fine
5. Se il valore cercato è maggiore, sposta avanti l'indice di inizio
6. Se il valore cercato è uguale, FINE

Pseudo - codice

Ricorsivo:

1. Controlla l'elemento in mezzo
2. Se il valore cercato è maggiore, richiama la funzione sulla seconda metà
3. Se il valore cercato è minore, richiama la funzione sulla prima metà
4. Se il valore cercato è uguale, FINE

Conclusioni

Ovviamente l'algoritmo di ricerca binaria è applicabile solo nel caso in cui l'array sia ordinato.

Se così non fosse, bisogna trovare il modo di ordinarlo, attraverso uno dei numerosi **Algoritmi di Ordinamento**.

Esercizi per casa

Implementare la ricerca binaria sia attraverso l'algoritmo ricorsivo sia attraverso quello iterativo.

