

Algoritmi di ricerca in linguaggio C

Ricerca sequenziale

```
#include <stdio.h>
#define MAX_DIM 5

/*
 * Funzione ricerca_seq(...)
 *
 * Cerca un numero all'interno di un vettore.
 * Restituisce la posizione del numero all'interno del vettore
 * oppure -1 se il numero non e' presente.
 *
 * Complessita' computazionale:
 *   BEST CASE: O(1)
 *   AVERAGE CASE: O(n)
 *   WORST CASE: O(n)
 */

int ricerca_seq(int vett[], int dim, int numero) {
    int i = 0;
    while (i < dim) {
        if (vett[i] == numero)
            return i;
        i++;
    }
    // Se il ciclo termina, il numero non e' presente nell'array
    return -1;
}

int main( void ) {

    int v[MAX_DIM] = {-30, 25, 30, 1, -10};
    int n, indice;

    printf("Inserire il numero da ricercare: ");
    scanf("%d", &n);

    indice = ricerca_seq(v, MAX_DIM, n);

    if (indice >= 0)
        printf("Il numero %d e' stato trovato in posizione %d.\n", n, indice);
    else
        printf("Il numero %d non e' presente nel vettore.\n", n);

    return 0;
}
```

Ricerca binaria (o dicotomica)

Ricerca binaria con algoritmo iterativo

```
#include <stdio.h>
#define MAX_DIM 5

/*
 * Funzione ricerca_bin(...)
 *
 * Cerca un numero all'interno di un vettore ordinato (metodo iterativo).
 * Restituisce la posizione del numero all'interno del vettore
 * oppure -1 se il numero non e' presente.
 *
 * PRECONDIZIONE: il vettore passato come parametro di input deve
 *                 essere gia' ordinato.
 *
 * Complessita' computazionale:
 *   BEST CASE: O(1)
 *   AVERAGE CASE: O(log(n))
 *   WORST CASE: O(log(n))
 */

int ricerca_bin(int vett[], int dim, int numero) {
    int i_inf = 0;
    int i_sup = dim - 1;
    int i_medio;

    while (i_inf <= i_sup) {
        i_medio = (i_inf + i_sup) / 2;
        if (numero > vett[i_medio])
            i_inf = i_medio + 1;
        else if (numero < vett[i_medio])
            i_sup = i_medio - 1;
        else
            return i_medio;        // Numero trovato ..
    }
    // Se il ciclo termina, il numero non e' presente nell'array
    return -1;
}

int main( void ) {

    int v[MAX_DIM] = {-30, 10, 1, 25, 30};    // Valori ordinati
    int n, indice;

    printf("Inserire il numero da ricercare: ");
    scanf("%d", &n);

    indice = ricerca_bin(v, MAX_DIM, n);

    if (indice >= 0)
        printf("Il numero %d e' stato trovato in posizione %d.\n", n, indice);
    else
        printf("Il numero %d non e' presente nel vettore.\n", n);

    return 0;
}
```

Ricerca binaria con algoritmo ricorsivo

```
#include <stdio.h>
#define MAX_DIM 5

/*
 * Funzione ricerca_bin_ric(...)
 *
 * Cerca un numero all'interno di un vettore ordinato (metodo ).
 * Restituisce la posizione del numero all'interno del vettore
 * oppure -1 se il numero non e' presente.
 *
 * PRECONDIZIONE: il vettore passato come parametro di input deve
 *                 essere gia' ordinato.
 *
 * Complessita' computazionale:
 *   BEST CASE: O(1)
 *   AVERAGE CASE: O(log(n))
 *   WORST CASE: O(log(n))
 */

int ricerca_bin_ric(int vett[], int i_inf, int i_sup, int numero) {
    int i_medio;

    if (i_inf > i_sup)
        return -1;

    i_medio = (i_inf + i_sup) / 2;

    if (vett[i_medio] == numero)
        return i_medio;

    if (numero > vett[i_medio])
        return ricerca_bin_ric(vett, i_medio+1, i_sup, numero);
    else
        return ricerca_bin_ric(vett, i_inf, i_medio-1, numero);
}

int main( void ) {

    int v[MAX_DIM] = {-30, 10, 1, 25, 30};    // Valori ordinati
    int n, indice;

    printf("Inserire il numero da ricercare: ");
    scanf("%d", &n);

    indice = ricerca_bin_ric(v, 0, MAX_DIM-1, n);

    if (indice >= 0)
        printf("Il numero %d e' stato trovato in posizione %d.\n", n, indice);
    else
        printf("Il numero %d non e' presente nel vettore.\n", n);

    return 0;
}
```