Linear Search in C++

how to sort an array in spp

Did you mean **SPP (Shortest Path Problem)** or was it a typo for **C++**? If you meant **C++**, you can use the built-in `sort()` function from the `<algorithm>` library.

Sorting an Array in C++

cpp

```cpp
#include <iostream>
#include <algorithm> // Required for sort()
using namespace std;

int main() {
    int arr[] = {50, 20, 40, 10, 30};
    int size = sizeof(arr) / sizeof(arr[0]);

    // Sorting the array in ascending order
    sort(arr, arr + size);

    // Display sorted array
    cout << "Sorted array: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

Sorting in Descending Order

cpp

```cpp
sort(arr, arr + size, greater<int>());
```

Let me know if you meant something else by **SPP**! 🚀

explaing sorting in descending order

Sorting in Descending Order in C++

In C++, we can use the `sort()` function from the `<algorithm>` library to sort an array. By default, it sorts in **ascending order**, but we can modify it to sort in **descending order** using `greater<int>()`.

Descending Order Sorting Example

```cpp
#include <iostream>
#include <algorithm> // Required for sort()
using namespace std;

int main() {
    int arr[] = {50, 20, 40, 10, 30};
    int size = sizeof(arr) / sizeof(arr[0]);

    // Sorting in descending order
    sort(arr, arr + size, greater<int>());

    // Display sorted array
    cout << "Sorted array in descending order: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

How It Works

1. `sort(arr, arr + size)` → Sorts the array in **ascending order**.
2. `greater<int>()` → A comparison function that tells `sort()` to arrange elements in **descending order**.
3. `arr, arr + size` → The range of elements to be sorted.

Output

```sql
Sorted array in descending order: 50 40 30 20 10
```

Would you like an explanation on how the `greater<int>()` function works internally? 🚀