Priority Queue:

- **Priority Queue:** Priority Queue ek qisam ka data structure hai jisme har data item apni priority ke mutabiq mojood hota hai.

Key Points:

1. **Tafseelat (Details):** Priority queue mein har item apni priority ke hisab se rakha jata hai. Jitni ziada priority, utna pehle.
2. **Amal (Operations):** Priority queue mein do basic amal hote hain:
    - **Daakhil Karna (Insertion):** Naya item dakhil karte waqt uski priority dekhi jati hai aur sahi jagah par rakh diya jata hai.
    - **Nikaalna (Deletion):** Sab se unchi priority wala item nikaala jata hai.
3. **Data Structure Type:** Priority queue ko aam tor par array ya linked list ke zariye implement kiya ja sakta hai.
4. **Misal (Example):**

```
class PriorityQ {
private int maxSize;
private long[] queArray;
private int nItems;

public PriorityQ(int s) {
    maxSize = s;
    queArray = new long[maxSize];
    nItems = 0;
}

public void insert(long item) {
    int j;
    if (nItems == 0)
        queArray[nItems++] = item;
    else {
            for (j = nItems - 1; j >= 0; j--) {
                if (item > queArray[j])
                    queArray[j + 1] = queArray[j];
                else
                    break;
            }
            queArray[j + 1] = item;
            nItems++;
        }
    }

    public long remove() {
        return queArray[--nItems];
    }

    public long peekMin() {
        return queArray[nItems - 1];
    }

    public boolean isEmpty() {
        return (nItems == 0);
    }

    public boolean isFull() {
        return (nItems == maxSize);
    }
}

class PriorityQApp {
    public static void main(String[] args) {
        int maxSize = 5;
        PriorityQ thePQ = new PriorityQ(maxSize);
        thePQ.insert(30);
        thePQ.insert(50);
        thePQ.insert(10);
        thePQ.insert(40);
        thePQ.insert(20);
        while (!thePQ.isEmpty()) {
            long item = thePQ.remove();
            System.out.print(item + " ");
        }
    }
}
```

Tafseelat (Details):

1. `PriorityQ` class mein `insert` function, naye item ko priority ke hisab se sahi jagah par dakhil karta hai.
2. `remove` function, sab se unchi priority wala item ko nikaalta hai.
3. `peekMin` function, sab se unchi priority wale item ko dekhta hai lekin nikaal nahi sakta.
4. `isEmpty` function, bataata hai ke queue khali hai ya nahi.
5. `isFull` function, bataata hai ke queue bhar gaya hai ya nahi.

Misal (Example):

```
int maxSize = 5;
PriorityQ thePQ = new PriorityQ(maxSize);
thePQ.insert(30);
thePQ.insert(50);
thePQ.insert(10);
thePQ.insert(40);
thePQ.insert(20);
```

```
while (!thePQ.isEmpty()) {
    long item = thePQ.remove();
    System.out.print(item + " ");
}
```

Chalne Ka Tariqa (Execution):

1. `PriorityQ` class ka object banaya jata hai.
2. `insert` function ke zariye items ko dakhil kiya jata hai.
3. `remove` function se items ko nikaala jata hai aur unko output diya jata hai.
4. Yehi amal jab tak chalta hai jab tak queue khali nahi hojati.

Kyun Use Karein (Why Use): Priority queue ka istemal tab kiya jata hai jab data items ko unki taeed ya tarteeb e nazar ke hisab se rakha jana zaruri hota hai, jaise ke job scheduling, graph algorithms, ya kisi bhi tareekh ki koshish mein.

Faida (Advantages): Priority queue ka istemal woh scenarios mein hota hai jahan data items ki taeed ya tarteeb e nazar ki zarurat hoti hai.

Mukhtasar Tafseelat (Summary): Priority queue ek aesa data structure hai jisme har item apni priority ke mutabiq mojood hota hai. Isme naye items ki priority dekhi jati hai aur unhe sahi tarteeb mein rakha jata hai. Java mein isey array ke zariye implement kiya ja sakta hai. Priority queue ki misal ke tor par, tafseelat ke sath diye gaye code ka istemal hota hai.

Yehi ek mukhtasar tohfa hai jo data items ko tarteeb dene mein madad karta hai.