```
class Solution {
    public String frequencySort(String s) {
        class Char {
            char c;
            int freq;
            public Char(char c, int freq) {
                this.c = c;
                this.freq = freq;
            }
        }
        Queue<Char> q = new PriorityQueue<>(
            new Comparator<Char>(){
                @Override
                public int compare(Char c1, Char c2) {
                    return c2.freq - c1.freq;
                }
            });

        Map<Character, Integer> freqMap = new HashMap<>();
        for (int i = 0; i < s.length(); i++) {
            int v = freqMap.getOrDefault(s.charAt(i), 0);
            freqMap.put(s.charAt(i), v+1);
        }
        for (Map.Entry<Character, Integer> entry : freqMap.entrySet()) {
            Char ch = new Char(entry.getKey(), entry.getValue());
            q.offer(ch);
        }
        StringBuilder sb = new StringBuilder();
        while (!q.isEmpty()) {
            Char ch = q.poll();
            for (int i = 0; i < ch.freq; i++) {
                sb.append(ch.c);
            }
        }
        return sb.toString();
    }
}
```