

最长的顺子

题目描述：

斗地主起源于湖北十堰房县，据传是一位叫吴修全的年轻人根据当地流行的扑克玩法“跑得快”改编的，如今已风靡整个中国，并流行于互联网上。

牌型：

单顺，又称顺子，最少 5 张牌，最多 12 张牌（3… A），不能有 2，也不能有大小王，不计花色

例如：3-4-5-6-7-8，7-8-9-10-J-Q，3-4-5-6-7-8-9-10-J-Q-K-A

可用的牌 3<4<5<6<7<8<9<10<J<Q<K<A<2 < B(小王)< C(大王)，每种牌除大小王外有 4 种花色(共有 13X4 + 2 张牌)

输入 1. 手上已有的牌 2. 已经出过的牌（包括对手出的和自己出的牌）

输出：对手可能构成的最长的顺子(如果有相同长度的顺子，输出牌面最大的那一个)，如果无法构成顺子，则输出 NO-CHAIN

输入描述：

输入的第一行为当前手中的牌

输入的第二行为已经出过的牌

输出描述：

最长的顺子

示例 1

输入：

3-3-3-3-4-4-5-5-6-7-8-9-10-J-Q-K-A

4-5-6-7-8-8-8

输出：

9-10-J-Q-K-A

说明：

示例 2

输入：

3-3-3-3-8-8-8-8

K-K-K-K

输出：

NO-CHAIN

说明：

剩余的牌无法构成顺子

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        String s = sc.nextLine();
```

```
        String p = sc.nextLine();
```

```
        Map<String, Integer> JtoEle = new HashMap<>();
```

```
        Map<Integer, String> eleToJ = new HashMap<>();
```

```
        String [] ch = new String[] {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J",  
        "Q", "K", "A"};
```

```
        int [] cnt = new int [15];
```

```
        for (int i = 3; i <= 14; i++) {
```

```
            cnt [i] = 4;
```

```
            JtoEle.put(ch[i], i );
```

```
            eleToJ.put(i, ch[i]);
```

```
        }
```

```
        for (String now : s.split("-")) {
```

```
            cnt[JtoEle.getOrDefault(now, 0)]--;
```

```
        }
```

```
        for (String now : p.split("-")) {
```

```
            cnt[JtoEle.getOrDefault(now, 0)]--;
```

```
        }
```

```
        int MaxLen = 0;
```

```

int from = 0;
for (int i = 3; i <= 10; i++) {
    if (cnt[i] == 0)continue;
    int j = i;
    while (j + 1 <= 14 && cnt[j + 1] > 0)j++;

    if (j - i + 1 >= MaxLen) {
        MaxLen = j - i + 1;
        from = i;
    }
}
if (MaxLen < 5) {
    System.out.println("NO-CHAIN");
    return;
}
for (int i = from; i < from + MaxLen; i++) {
    if (i != from) System.out.print("-");
    System.out.print(eleToj.get(i));
}
}
}

```