第一题 剑指 offer 原题 最大子序列和

```
1 package  便利蜂;
2
3 import  java.util.Scanner;
4
5 public  class  子序列最大和  {
6
7         public  static  void  main(String[]  args)  {
8                 //  TODO  Auto-generated  method  stub
9                 Scanner  sc  =  new  Scanner(System.in);
10                String[]  arr  =  sc.nextLine().split(",");
11                System.out.println(maxSum(arr));
12        }
13
14        private  static  int  maxSum(String[]  arr)  {
15                int  sum  =  0;
16                int  res  =  0;
17                for(String  str:  arr)  {
18                        int  num  =  Integer.valueOf(str);
19                        sum  =  Math.max(num,  sum+num);
20                        res  =  Math.max(sum,  res);
21                }
22                return  res;
23        }
24
25}
```

第二题 半小时做了个有向图 用个 HashMap 保存节点映射方便查询 DFS 获得路径放进 ArrayList 里 最后打印出来

```
1 package  便利蜂;
2
3 import  java.util.ArrayList;
4 import  java.util.HashMap;
5 import  java.util.LinkedList;
6 import  java.util.Scanner;
7
8 public  class  高老庄  {
9
10        private  HashMap<String,  Address>  map  =  new  HashMap()
11;;
12        private  ArrayList<Address>  res;
13
14        class  Address{
15                private  String  addressName;
```

```
16                private  boolean  isReached  =  false;
17                private  Address  east;
18                private  Address  west;
19                private  Address  north;
20                private  Address  south;
21                public  Address(String  addressName)  {
22                        this.addressName  =  addressName;
23                }
24        }
25
26
27        public  static  void  main(String[]  args)  {
28
29                Scanner  sc  =  new  Scanner(System.in);
30                String[]  arr  =  sc.nextLine().split(",");
31
32                高老庄  main  =  new  高老庄();
33                main.generateMap();
34                main.findPath(arr[0],  arr[1]);
35
36                main.printPath();
37        }
38
39        private  void  printPath()  {
40                for(int  i=1;  i<res.size();  i++)  {
41                        Address  cur  =  res.get(i);
42                        Address  pre  =  res.get(i-1);
43                        if(pre.north  ==  cur)  System.out.print("
44north");
45                        else  if(pre.south  ==  cur)  System.out.p
46rint("south");
47                        else  if(pre.west  ==  cur)  System.out.pr
48int("west");
49                        else  if(pre.east  ==  cur)  System.out.pr
50int("east");
51                        if(i  !=  res.size()-1)  System.out.print(
52",");
53                }
54        }
55
56        public  void  findPath(String  wukong,  String  bajie)  {
57                Address  start  =  map.get(wukong);
58                Address  end  =  map.get(bajie);
59                ArrayList<Address>  path  =  new  ArrayList();
```

```
60                path.add(start);
61                start.isReached = true;
62                dfs(start, end, path);
63        }
64
65        private void dfs(Address start, Address end, ArrayLi
66st<Address> path) {
67
68                if(start == end) {
69                        if(res == null || path.size() < res.
70size()) {
71                                res = new ArrayList(path);
72                        }
73                        return;
74                }
75                int isEnd = 0;
76
77                if(start.north != null && !start.north.isReach
78ed) {
79                        path.add(start.north);
80                        start.north.isReached = true;
81                        dfs(start.north, end, path);
82                        path.remove(path.size()-1);
83                        start.north.isReached = false;
84                }
85                if(start.south != null && !start.south.isReach
86ed) {
87                        start.south.isReached = true;
88                        path.add(start.south);
89                        dfs(start.south, end, path);
90                        path.remove(path.size()-1);
91                        start.south.isReached = false;
92                }
93                if(start.west != null && !start.west.isReached
94) {
95                        start.west.isReached = true;
96                        path.add(start.west);
97                        dfs(start.west, end, path);
98                        path.remove(path.size()-1);
99                        start.west.isReached = false;
10                }
0                 if(start.east != null && !start.east.isReached
10) {
1                         start.east.isReached = true;
```

```
10                      path.add(start.east);
2                       dfs(start.east,  end,  path);
10                      path.remove(path.size()-1);
3                       start.east.isReached  =  false;
10              }
4
10      }
5
10      public  void  generateMap()  {
6               Address  DAOTIAN1  =  new  Address("DAOTIAN1");
10              Address  TULU1  =  new  Address("TULU1");
7               Address  DAOTIAN  =  new  Address("DAOTIAN");
10              Address  TULU  =  new  Address("TULU");
8               Address  CUNKOU  =  new  Address("CUNKOU");
10              Address  NONGSHE  =  new  Address("NONGSHE");
9               Address  TULU2  =  new  Address("TULU2");
11              Address  JIEDAO  =  new  Address("JIEDAO");
0               Address  TIEPU  =  new  Address("TIEPU");
11              Address  LIUJIABUDIAN  =  new  Address("LIUJIABUDI
1 AN");
11              Address  JIEDAO1  =  new  Address("JIEDAO1");
2               Address  XIAOJIUGUAN  =  new  Address("XIAOJIUGUAN
11");
3               Address  ZHANGFANG  =  new  Address("ZHANGFANG");
11              Address  PIANTING  =  new  Address("PIANTING");
4               Address  GUIGE  =  new  Address("GUIGE");
11              Address  YASHI  =  new  Address("YASHI");
5               Address  HUAYUAN  =  new  Address("HUAYUAN");
11              Address  HOUYUAN  =  new  Address("HOUYUAN");
6               Address  ZHENGTING  =  new  Address("ZHENGTING");
11              Address  ZHENGYUAN  =  new  Address("ZHENGYUAN");
7               Address  GAOJIADAYUAN  =  new  Address("GAOJIADAYU
11AN");
8               Address  JIEDAO2  =  new  Address("JIEDAO2");
11              Address  PIANFANG  =  new  Address("PIANFANG");
9               Address  FANTING  =  new  Address("FANTING");
12              Address  XIYIFANG  =  new  Address("XIYIFANG");
0               Address  TULU3  =  new  Address("TULU3");
12              Address  QINGSHILU  =  new  Address("QINGSHILU");
1
12              DAOTIAN1.south  =  TULU1;
2               TULU1.north  =  DAOTIAN1;
12              TULU1.south  =  DAOTIAN;
3               DAOTIAN.north  =  TULU1;
```

```
12                  DAOTIAN.south  =  TULU;
4                   TULU.north  =  DAOTIAN;
12                  TULU.south  =  CUNKOU;
5                   CUNKOU.north  =  TULU;
12                  CUNKOU.east  =  NONGSHE;
6                   NONGSHE.west  =  CUNKOU;
12                  TULU1.east  =  TULU2;
7                   TULU2.west  =  TULU1;
12                  TULU2.east  =  JIEDAO;
8                   JIEDAO.west  =  TULU2;
12                  LIUJIABUDIAN.south  =  JIEDAO;
9                   JIEDAO.north  =  LIUJIABUDIAN;
13                  JIEDAO.south  =  TIEPU;
0                   TIEPU.north  =  JIEDAO;
13                  JIEDAO.east  =  JIEDAO1;
1                   JIEDAO1.west  =  JIEDAO;
13                  JIEDAO1.south  =  XIAOJIUGUAN;
2                   XIAOJIUGUAN.north  =  JIEDAO1;
13                  YASHI.south  =  GUIGE;
3                   GUIGE.north  =  YASHI;
13                  GUIGE.east  =  HOUYUAN;
4                   HOUYUAN.west  =  GUIGE;
13                  PIANTING.east  =  ZHENGTING;
5                   ZHENGTING.west  =  PIANTING;
13                  ZHANGFANG.east  =  ZHENGYUAN;
6                   ZHENGYUAN.west  =  ZHANGFANG;
13                  JIEDAO1.east  =  GAOJIADAYUAN;
7                   GAOJIADAYUAN.west  =  JIEDAO1;
13                  HUAYUAN.south  =  HOUYUAN;
8                   HOUYUAN.north  =  HUAYUAN;
13                  HOUYUAN.south  =  ZHENGTING;
9                   ZHENGTING.north  =  HOUYUAN;
14                  ZHENGYUAN.north  =  ZHENGTING;
0                   ZHENGTING.south  =  ZHENGYUAN;
14                  ZHENGYUAN.south  =  GAOJIADAYUAN;
1                   GAOJIADAYUAN.north  =  ZHENGYUAN;
14                  HOUYUAN.east  =  XIYIFANG;
2                   XIYIFANG.west  =  HOUYUAN;
14                  ZHENGTING.east  =  FANTING;
3                   FANTING.west  =  ZHENGTING;
14                  ZHENGYUAN.east  =  PIANFANG;
4                   PIANFANG.west  =  ZHENGYUAN;
14                  GAOJIADAYUAN.east  =  JIEDAO2;
5                   JIEDAO2.west  =  GAOJIADAYUAN;
```

```
14                JIEDAO2.east  =  TULU3;
6                 TULU3.west  =  JIEDAO2;
14                TULU3.east  =  QINGSHILU;
7                 QINGSHILU.west  =  TULU3;
14
8
14                map.put("DAOTIAN1",  DAOTIAN1);
9                 map.put("TULU1",  TULU1);
15                map.put("DAOTIAN",  DAOTIAN);
0                 map.put("TULU",  TULU);
15                map.put("CUNKOU",  CUNKOU);
1                 map.put("NONGSHE",  NONGSHE);
15                map.put("TULU2",  TULU2);
2                 map.put("LIUJIABUDIAN",  LIUJIABUDIAN);
15                map.put("JIEDAO",  JIEDAO);
3                 map.put("TIEPU",  TIEPU);
15                map.put("YASHI",  YASHI);
4                 map.put("GUIGE",  GUIGE);
15                map.put("PIANTING",  PIANTING);
5                 map.put("ZHANGFANG",  ZHANGFANG);
15                map.put("JIEDAO1",  JIEDAO1);
6                 map.put("XIAOJIUGUAN",  XIAOJIUGUAN);
15                map.put("HUAYUAN",  HUAYUAN);
7                 map.put("HOUYUAN",  HOUYUAN);
15                map.put("ZHENGTING",  ZHENGTING);
8                 map.put("ZHENGYUAN",  ZHENGYUAN);
15                map.put("GAOJIADAYUAN",  GAOJIADAYUAN);
9                 map.put("XIYIFANG",  XIYIFANG);
16                map.put("FANTING",  FANTING);
0                 map.put("PIANFANG",  PIANFANG);
16                map.put("JIEDAO2",  JIEDAO2);
1                 map.put("TULU3",  TULU3);
16                map.put("QINGSHILU",  QINGSHILU);
2         }
16}
```

第三题 简单的 DFS 吧，但是我觉得应该有更好的方法，DFS 还是有点太蠢了

```
1 package  便利蜂;
2
3 import  java.util.Scanner;
4
5 public  class  运送货物  {
6
```

```
7       private static int res = Integer.MAX_VALUE;
8
9       public static void main(String[] args) {
1               Scanner sc = new Scanner(System.in);
0               String[] arr = sc.nextLine().split(",");
1               dfs(arr, 0, 0);
1               if(res == Integer.MAX_VALUE){
1                       System.out.println(-1);
2               }else{
1                       System.out.println(res);
3               }
1       }
4
1       private static void dfs(String[] arr, int index, int
5  times) {
1               if(index + 1 > arr.length) return;
6               if(index + 1 == arr.length) {
1                       res = times < res ? times: res;
7                       return;
1               }
8               int oil = Integer.valueOf(arr[index]);
1               for(int i=index+1; i <= index+oil; i++) {
9                       times++;
2                       dfs(arr, i, times);
0                       times--;
2               }
1       }
```