

[首页](#) [新闻](#) [论坛](#) [问答](#) [博客](#) [招聘](#) [更多](#) ▼  
[您还未登录!](#) [我的应用](#) [登录](#) [注册](#)

## 曲径通幽处

永久域名 <http://xiaobo.javaeye.com>  
[5个人30秒过河问题\(原创\)](#) | [Oracle SYS\\_CONTEXT Function](#)  
2010-04-05

### [三个僧侣过河问题模拟实现\(原创\)](#)

#### 文章分类:Java编程

今天突然想起以前同事提到过的一个过河问题，游戏设计思路

需要使用程序代码体现其过程，心血来潮，来一把.....如下：

三个僧侣和三个食人族过河

目的：需要把他们六个一起使用一条船送到河的对面岸上

条件：当食人族的数目超出僧侣的数目时，就会出现食人族吃人的想象(是不允许的)

每条船最多能坐下两个(僧侣或者食人族),每次船启动时必须有一个在船上。

思路：必须先所有的食人族过河，再利用食人族将所有的僧侣送过河，最后将食人族再送过河。

模拟僧侣的攻击力为4(总攻击力为12)，食人族攻击力为3(总攻击力为9)，

当每边食人族的攻击力和超出僧侣的攻击力和的时候就发生事故，

必须控制僧侣攻击力和超出食人族攻击力和的值。

结果：最短应该是小于等于11次往返次数

执行环境(WinXP+java version "1.5.0\_06")

#### Java代码

```
1. /**
2.  *created by zxb
3.  *date 2010-4-5 - 上午10:05:38
4.  *zxb 开源测试项目 test
5.  *to do TODO
6.  **/
7.
8.
9. package com.java.algorism;
10.
11. import java.util.ArrayList;
12. import java.util.Collections;
13. import java.util.List;
14. import java.util.Map;
15. import java.util.Random;
16.
17. /**
18.  * 三个僧侣和三个食人族过河
19.  * 目的：需要把他们六个一起使用一条船送到河的对面岸上
20.  * 条件：当食人族的数目超出僧侣的数目时，就会出现食人族吃人的想象(是不允许的)
21.  * 每条船最多能坐下两个(僧侣或者食人族),每次船启动时必须有一个在船上。
22.  * 思路：必须先所有的食人族过河，再利用食人族将所有的僧侣送过河，最后将食人族再送过河。
```

```

23. * 模拟僧侣的攻击力为4(总攻击力为12), 食人族攻击力为3(总攻击力为9),
24. * 当每边食人族的攻击力和超出僧侣的攻击力和的时候就发生事故,
25. * 必须控制僧侣攻击力和超出食人族攻击力和的值。
26. * 结果: 最短应该是小于等于11次往返次数
27. */
28. public class CrossRiver {
29.
30.     private static final List<Integer> list=new ArrayList<Integer>();
31.     /**
32.      * @param args
33.      */
34.     public static void main(String[] args) {
35.         Senglv s1=new Senglv();
36.         Senglv s2=new Senglv();
37.         Senglv s3=new Senglv();
38.         Shirenu sr1=new Shirenu();
39.         Shirenu sr2=new Shirenu();
40.         Shirenu sr3=new Shirenu();
41.         List<Object> l1=new ArrayList<Object>();
42.         List<Object> l2=new ArrayList<Object>();
43.         l1.add(s1);l1.add(s2);l1.add(s3);
44.         l2.add(sr1);l2.add(sr2);l2.add(sr3);
45.         List<Object> right=new ArrayList<Object>();//右边数据
46.         List<Object> left=new ArrayList<Object>();//左边数据
47.         right.addAll(l1);right.addAll(l2);
48.         for (int i = 1; i < 1000; i++) {
49.             boolean flag=((i+1)%2)==0?true:false;
50.             transportData(flag?left:right,flag?right:left,flag);
51.         }
52.         Collections.sort(list);
53.         System.out.println("最小折返次数: "+list.get(0));
54.     }
55.
56.     /**
57.      *
58.      * @param left
59.      * @param right
60.      * @param ship=true 为右边;ship=false 为左边
61.      */
62.     public static void transportData(List<Object> left,List<Object> right,boolean ship){
63.         int count=0;//计数器
64.         int k=0;
65.         List<Object> l=null;
66.         while((right !=null && (k=right.size())>0)){
67.             if(ship){
68.                 l=getRandomNum(right,2);
69.                 left.addAll(l);
70.                 right.removeAll(l);
71.                 if(!compareATK(right) || !compareATK(left)){
72.                     right.addAll(l);
73.                     left.removeAll(l);
74.                     continue;
75.                 }
76.                 ship=false;
77.                 count++;
78.             }else{
79.                 l=getRandomNum(left,2);
80.                 left.removeAll(l);

```

```

81.         right.addAll(l);
82.         if(!compareATK(right) || !compareATK(left)){
83.             right.removeAll(l);
84.             left.addAll(l);
85.             continue;
86.         }
87.         ship=true;
88.         count++;
89.     }
90. }
91. //System.out.println("往返次数==="+count);
92. list.add(count);
93. }
94.
95. /**
96.  * 攻击力比较大小 僧侣的攻击力和大于食人族攻击力和时返回true,
97.  * 当一边为空时也返回true,否则计算实际攻击力大小
98.  * @param l
99.  * @return
100. */
101. public static boolean compareATK(List<Object> l){
102.     int s1=0,s2=0;
103.     for (int i = 0; i < l.size(); i++) {
104.         if(l.get(i) instanceof Senglv){
105.             s1+=((Senglv)l.get(i)).getATK();
106.         }else if(l.get(i) instanceof Shirenzuo){
107.             s2+=((Shirenzuo)l.get(i)).getATK();
108.         }
109.     }
110.     return s1==0 || s2==0 || s1>s2;
111. }
112. /**
113.  * 从List集合中随机抽取count个数以下的list集合(list.size()<=count)
114.  * @param list
115.  * @param count
116.  * @return
117.  */
118. public static List<Object> getRandomNum(List<Object> list,int count){
119.     List<Object> l=new ArrayList<Object>();
120.     int k=0,m=0;
121.     k=(list!=null?list.size():0);
122.     if(list!=null && k>=count){
123.         Collections.shuffle(list);
124.         m=(new Random().nextInt(count)+1);
125.         for (int i = 0; i < m; i++) {
126.             l.add(list.get(i));
127.         }
128.     }else{
129.         l.addAll(list);
130.     }
131.     return l;
132. }
133.
134. /**
135.  *
136.  * @param m1
137.  * @param m2
138.  * @return
139.  */

```

```
140. public static boolean compareATK(Map<String,Integer> m1,Map<String,Integer> m2){
141.     int s1=0,s2=0;
142.     while(m1.values().iterator().hasNext()){
143.         s1+=m1.values().iterator().next();
144.     }
145.     while(m2.values().iterator().hasNext()){
146.         s2+=m2.values().iterator().next();
147.     }
148.     return s1>s2;
149. }
150. }
151.
152. class Senglv{
153.
154.     public Senglv(){
155.     }
156.
157.     private int ATK=4;
158.
159.     public int getATK() {
160.         return ATK;
161.     }
162.
163.     public void setATK(int atk) {
164.         ATK = atk;
165.     }
166. }
167.
168. class Shirenzuo{
169.
170.     public Shirenzuo(){
171.     }
172.
173.     private int ATK=3;
174.
175.     public int getATK() {
176.         return ATK;
177.     }
178.
179.     public void setATK(int atk) {
180.         ATK = atk;
181.     }
182. }
```

[電玩遊戲大廠大舉徵才](#)

Blizzard/智凡迪/中華網龍等知名廠商 企劃、行銷、軟體/測試工程師...等  
[www.1111.com.tw](http://www.1111.com.tw)

[5个人30秒过河问题\(原创\)](#) | [Oracle SYS\\_CONTEXT Function](#)

- 16:10
- 浏览 (155)
- [评论](#) (0)
- 分类: [JAVA](#)
- [相关推荐](#)

评论

发表评论

表情图标



B I U Quote Code List Img URL Flash Table

字体颜色: 标准 字体大小: 标准 对齐: 标准


提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

Large empty text area for user input.

您还没有登录, 请[登录](#)后发表评论(快捷键 Alt+S / Ctrl+Enter)



oboaix

- 浏览: 72451 次
-  我现在离线
- [详细资料](#) [留言簿](#)

搜索本博客

Search input field with a '搜索' button.

最近访客 [>>更多访客](#)



[bianmao](#)



[liweixw](#)



[li-haitao](#)

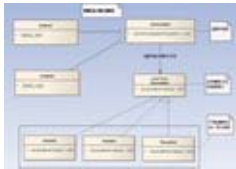


[humanbeing999](#)

### 博客分类

- [全部博客 \(80\)](#)
- [Database \(28\)](#)
- [JAVA \(20\)](#)
- [java basic \(4\)](#)
- [javascript \(7\)](#)
- [music 文章 \(6\)](#)
- [体育 \(6\)](#)
- [转载 \(10\)](#)
- [得失寸草心 \(4\)](#)
- [E文 \(1\)](#)

### 我的相册



strategy

[共 4 张](#)

### 我的留言簿 [>>更多留言](#)

- 好厉害 我是杨熙 怎么加你为好友?  
-- by [xiyangzk](#)

### 其他分类

- [我的收藏 \(0\)](#)
- [我的论坛主题贴 \(1\)](#)
- [我的所有论坛贴 \(10\)](#)
- [我的精华良好贴 \(0\)](#)




### 最近加入圈子

- [珠海之夜](#)
- [篮球俱乐部](#)
- [英语学习](#)

### 存档

- [2010-06 \(1\)](#)
- [2010-04 \(2\)](#)
- [2010-03 \(1\)](#)
- [更多存档...](#)

### 评论排行榜

- [Oracle错误: ORA-00604处理](#)
- [delete,truncate,drop 区别\(ORACLE\)](#)
- [ORACLE常用SQL优化hint语句](#)
- [5个人30秒过河问题\(原创\)](#)
- [Rman学习笔记记录](#)
-  [RSS](#)
-  [Add to Google](#)
-  [订阅到 鲜果](#)
- [\[什么是RSS?\]](#)

---

声明: JavaEye文章版权属于作者, 受法律保护。没有作者书面许可不得转载。  
若作者同意转载, 必须以超链接形式标明文章原始出处和作者。

© 2003-2010 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [ 沪  
ICP备05023328号 ]