

数据结构 A

作业 5 参考答案

作业情况

教材：数据结构教程（C++ 语言描述）李春葆等

题目范围：队列

邮箱：wjyyy1@126.com

授课教师：彭蓉 教授

助教：王骏骁

习题 1 单选题

已知环形队列存储在一维数组 $A[0..n-1]$ 中，且队列非空时 $front$ 和 $rear$ 分别指向队头元素和队尾元素。若初始时队列空，且要求第一个进入队列的元素存储在 $A[0]$ 处，则初始时 $front$ 和 $rear$ 的值分别是（ ）

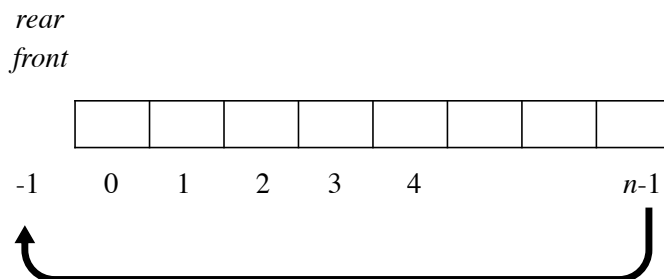
- A. 0, 0
- B. 0, $n-1$
- C. $n-1$, 0
- D. $n-1$, $n-1$

答案：B

解析：题目要求：若初始时队列空，且要求第一个进入队列的元素存储在 $A[0]$ 处。

而参考书上对队列的定义， $front$ 指向队首元素的前一个位置，队尾指针 $rear$ 指向队尾元素。本题要求 $front$ 指向队首元素，队尾指针 $rear$ 指向队尾元素。所以初始状态下应有 $front = rear + 1$ 。

队列的下一个元素会被插入队尾 $rear+1$ 的位置，因此可以确定 $rear = n-1$ 。又因为队列为空，所以 $front = rear+1$ 。此时循环需要对 $MaxSize$ 取余，所以 $front = 0$ 。



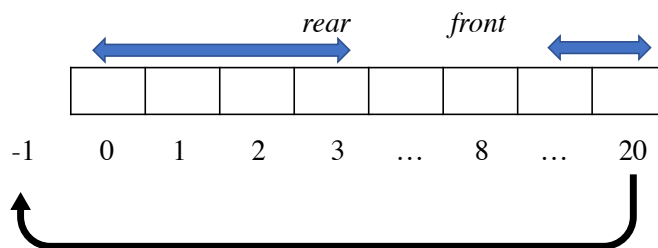
习题 2 单选题

设环形队列的存储空间是 $a[0..20]$ ，且当前队头指针 (f 指向队首元素的前一个位置) 和队尾指针 (r 指向队尾元素) 的值分别是 8 和 3，则该队列中的元素个数为 ()

- A. 5
- B. 6
- C. 16
- D. 17

答案: C

解析: $f = 8, r = 3$ ，从 f 的下一个位置开始排布元素，直到 r 的位置，因此 $[9, 20]$ 和 $[0, 3]$ 都是有值的。一共为 $(20 - 9 + 1) + (3 - 0 + 1) = 16$ 个。



习题 3 单选题

栈和队列的共同特点是 ()。

- A. 都是先进后出
- B. 都是先进先出
- C. 只允许在端点处插入和删除元素
- D. 没有共同点

答案: C

解析: 栈是先进后出，队列是先进先出。在脑海里模拟两个管道，其中一个是一边出入的，另一边被封起来；另一个是一边入一边出的。

习题 4 填空题

循环队列 $A[0, 10]$ ，存储容量是 10，队首指针 f ，队尾指针 r ，初始状态 ()。假设经过一系列入队和出队操作后， $r = f = 5$ ，然后又出队 1 个元素，则现在循环队列中的元素个数为 ()

答案: $r = f$, 9 或 0 (下溢错误)

解析：第二问答案给的是 9 或 0，我认为 9 更对一些，只填 9 的同学我给了满分，而只填 0 的同学我扣掉了一分。

首先初始状态应该填的是 f, r 两变量的状态。要注意本题是循环队列，初始状态下 $f = r$ ，默认赋值为 0。因此写 $f = r$ 、 $f = 0, r = 0$ 的同学都算对。

$r = f = 5$ 时，考虑两种情况：队空或队满（恰好 10 个元素）。如果队满，又出队 1 个元素则剩下 9 个。如果队空，严格意义上不能又出队 1 个元素，但是如果一定要再出的话那就仍然是空队列。

因此本题第二个空填 9 或 0。如果要填 0 可以标注“下溢错误”。

习题 5 团队队列

【问题描述】

在团队队列中每个成员都属于一个团队，如果一个成员进入队列，它首先从头到尾搜索队列，以检查它的一些队友（同一队的成员）是否已经在队列中，如果是，它会进入到该团队的后面，如果不是，它会从尾部进入队列并成为新的最后一个成员。成员出队是按常规队列操作，按照出现在队列中的顺序从头到尾进行处理。你的任务是编写一个模拟这样的团队队列的程序。

【输入形式】

每个测试用例都以团队个数 t 开始 ($1 \leq t \leq 1000$)，然后是团队描述，每个描述包含属于团队的成员个数和成员编号列表，成员编号为 0 到 999999 之间的整数，一个团队最多可以包含 1000 个成员。然后是一系列命令，有三种不同的命令：

1. ENQUEUE p ：成员 p 进入队列。
2. DEQUEUE：队列中第一个成员出来并将其从队列中删除。
3. STOP：当前测试用例结束。

【输出形式】

对于每个 DEQUEUE 命令，以单独一行输出出队的成员。

【样例输入】

```
2
3 101 102 103
3 201 202 203
ENQUEUE 101
ENQUEUE 201
ENQUEUE 102
ENQUEUE 202
ENQUEUE 103
```

```
ENQUEUE 203
```

```
DEQUEUE
```

```
DEQUEUE
```

```
DEQUEUE
```

```
DEQUEUE
```

```
DEQUEUE
```

```
DEQUEUE
```

```
STOP
```

【样例输出】

```
101
```

```
102
```

```
103
```

```
201
```

```
202
```

```
203
```

【样例说明】

测试数据的文件名为 `in.txt`。

【评分标准】

该题目有 10 个测试用例，每通过一个测试用例，得 10 分。

答案：

```
1 #include<iostream>
2 #include<fstream>
3 #include<queue>
4 using namespace std;
5 queue<int> Q;
6 queue<int> q[1010];
7 int team[1000000];
8 //team[i]表示i属于哪个队伍
9 int main()
10 {
11     ifstream in("in.txt");
12     string op;
13     int t;
14     in>>t;
15     for(int i=1;i<=t;i++)
16     {
17         int n,x;
18         in>>n;
```

```
19     for(int j=1;j<=n;j++)
20     {
21         in>>x;
22         team[x]=i;
23     }
24 }
25 in>>op;
26 while(op!="STOP")
27 {
28     if(op=="ENQUEUE")
29     {
30         int x;
31         in>>x;
32         if(q[team[x]].empty())
33             //或写成 tt=team[x] 并判断 q[tt] 是否为空也可以
34             Q.push(team[x]);
35         q[team[x]].push(x);
36     }
37     else// 否则就是DEQUEUE
38     {
39         //DEQUEUE的内容一定是Q里第一个团队的第一个元素
40         cout<<q[Q.front()].front()<<endl;
41         q[Q.front()].pop();
42         if(q[Q.front()].empty())//如果这个团队没有元素了,就出队
43             Q.pop();
44     }
45     in>>op;
46 }
47 return 0;
48 }
```

解析：本题更适合用链表写，插入元素时可以依次从前到后找到队列中是否有相同团队的元素，如果有则插入到相同团队的最后，否则插入到队列尾部。

但是因为团队之间不会相互干扰，所以可以将每个团队分开进行排列，并额外使用一个队列 Q 存储团队之间的顺序。每次插入成员时，先检查所属团队队列中是否有元素，如果有则直接插入到团队队列的最后，否则将团队插入到额外队列尾部，再将元素插入团队队列。

当一个团队队列被清空时，它也要被从额外队列中被移除了。

习题 6 扔钉子**【问题描述】**

年度学校自行车比赛开始了，ZL 是这所学校的学生，他太无聊了，因为他不能骑自行车！因此，他决定干预比赛，他通过以前的比赛视频获得了选手的信息，一个选手第一秒可以跑 F 米，然后每秒跑 S 米。每个选手有一条直线跑道，ZL 每秒向跑的最远的运动员跑道扔一个钉子，在自行车胎爆炸之后，该选手将被淘汰。如果有多个选手是 NO.1，则他总是选择 ID 最小的选手扔钉子。

【输入形式】

每个测试用例的第一行包含一个整数 $n(1 \leq n \leq 50000)$ ，表示选手人数，然后跟随 n 行，每行包含第 i 个选手的两个整数 $F_i(0 \leq F_i \leq 500)$ ， $S_i(0 < S_i \leq 100)$ ，表示该选手第一秒可以跑 F_i 米，然后每秒跑 S_i 米， i 是玩家从 1 开始的 ID 。

【输出形式】

输出 n 个数字，以空格分隔，第 i 个数字是选手的 ID ，该选手将在第 i 秒结束时被淘汰。

【样例输入】

```
3
100 1
100 2
3 100
```

【样例输出】

```
1 3 2
```

【样例说明】

测试数据的文件名为 in.txt。

【评分标准】

该题目有 10 个测试用例，每通过一个测试用例，得 10 分。

答案：

```
1 #include<iostream>
2 #include<fstream>
3 #include<vector>
4 #include<queue>
5 using namespace std;
6 struct player
7 {
8     int start,id;
9     friend bool operator <(player a,player b)
10    {
```

```
11         if(a.start!=b.start)//如果可以通过速度区分,就返回速度大小关系
12             return a.start<b.start;
13         return a.id>b.id;//如果速度相等,就返回id的反向关系
14     }
15 };
16 priority_queue<player> q[105];//q[i]包含了速度为i的所有人
17 int main()
18 {
19     ifstream in("in.txt");
20     int n;
21     in>>n;
22     for(int i=1;i<=n;i++)
23     {
24         player x;
25         int speed;
26         in>>x.start>>speed;
27         x.id=i;
28         q[speed].push(x);
29     }
30     for(int i=0;i<n;i++)//i=0表示第1秒。第i+1秒时每个人的位置是f+s*i
31     {
32         int farthest_j=0;
33         int dis=0;
34         for(int j=1;j<=100;j++)
35             if(!q[j].empty())
36             {
37                 player now=q[j].top();
38                 if(now.start+i*j>dis)
39                 {
40                     dis=now.start+i*j;
41                     farthest_j=j;
42                 }
43                 else if(now.start+i*j==dis)
44                 { //距离相等,则比较id
45                     if(now.id<q[farthest_j].top().id)
46                         farthest_j=j;
47                 }
48             }
49         //最后farthest_j队伍里跑得最远的会被淘汰
50         cout<<q[farthest_j].top().id<<" ";
51         q[farthest_j].pop();
52     }
53     return 0;
54 }
```

解析: 根据题意, S_i 的取值只有 100 种。因此 S_i 相同的人的前后位置是相对固定的, 淘汰顺序也因此固定。所以可以将 S_i 相同的选手放入同一个排好序的队列 q_{S_i} ,

可以使用优先队列 `priority_queue`。

每次被淘汰的人一定是从所有队列 q 的最大值中再选出它们的最大值。所以每次将所有优先队列的最大值进行比较，把最大的弹出。

举例理解一下就是现在有 100 个已经从前到后排好序的队伍，我们要找所有人里面最靠前的一个，那么这一个人一定在它自己队列里也是最靠前的。因此我们只需要每次找出所有队伍里面最靠前的一个，当把它删掉（弹出优先队列）时，它后面的元素会自动补位，而优先队列就可以做到这一点。

此处涉及两个关键字的比较，可以用结构体重载运算符来解决，也可以使用 `pair`，用法可以参考程序设计课程。如果对代码细节有问题可以咨询助教。

总结

题目：队列

日期：2024 年 4 月 21 日

批改人：王骏峤

邮箱：wjyyy1@126.com

习题 1：注意对概念的理解，相信大家都能够掌握队列的用法，但是如果考试考概念的话还是要注重细节。感谢黄冰豪同学对本题解析的指正。

习题 2：注意存储空间最末元素具体是哪一个，同时也要注意循环队列和普通队列的区别。

习题 3：同样是对概念的理解，如果这样的性质题目想不清楚就多画图来帮助理解。

习题 4：填空题只有写了我才可能想办法给大家一点思考分，不写的话我是无法给分的。只填 9 的同学可以认为是思考过循环队列的，这题可以得分。填空题看出有思考的同学我会酌情加分。（但是好像跟整次作业比起来，一分的差距不是很大）

习题 5：直接对题目模拟是一种做法，但是可能有超时的风险。如果能用更聪明更简单的做法达到目的，更鼓励大家尝试简便做法。在作业截止前提交次数是不受限制的，可以多次提交来验证自己的不同思路。

习题 6：此题和 <https://www.luogu.com.cn/problem/U31965> 有很相似的地方，有余力的同学可以考虑做一下并体会。题目难度有一点点大，需要多思考并和之前学过的内容进行结合。

各位同学如有问题欢迎及时在群里提出，或者通过邮件/QQ 联系我。