华北电力大学

**实 验 报 告**

|

|

**实验名称** 栈与递归

**课程名称**  数据结构综合实践

|

|

专业班级： 信息2202 学生姓名：徐梓乔

学 号： 220221100327 成 绩：

|  |
| --- |
| 一、实验内容  假设一列货运列车共有 n 节车厢，每节车厢将停放在不同的车站。假定 n 个车站的编号分别为1至  n ，货运列车按照第 n 站至第 1 站的次序经过这些车站。车厢的编号与它们的目的地相同。为了便  于从列车上卸掉相应的车厢，必须重新排列车厢，使各车厢从前至后按编号1至 n 的次序排列。当所  有的车厢都按照这种次序排列时，在每个车站只需卸掉最后一节车厢即可。  我们在一个转轨站里完成车厢的重排工作，在转轨站中有一个入轨、一个出轨和 k 个缓冲铁轨（位于  入轨和出轨之间）。图1（a）给出一个转轨站，其中有 k 个（ k=3 ）缓冲铁轨 H1 ， H2 和 H3 。开始  时， n 节车厢的货车从入轨处进入转轨站，转轨结束时各车厢从右到左按照编号1至 n 的次序离开转  轨站（通过出轨处）。  二、实验方法与步骤  //若需出轨的编号恰好是入轨的编号, 则直接出轨.  static bool Input(int c, ref int minH, ref int minS, LinkStack<int>[] h)  {  int bestTrack = -1;//目前最优的铁轨  int bestTop = int.MaxValue;//最优铁轨上的头辆车厢  for(int i = 0; i < h.Length; i++)  {  if (h[i].IsEmpty() == false)  {  int x = h[i].StackTop;  if (c < x && x < bestTop)  {  bestTop = x;  bestTrack = i;  }  }  else  {  if(bestTrack == -1)  {  bestTrack = i;  break;  }  }  }  if(bestTrack == -1)  {  return false;  }  h[bestTrack].Push(c);  Console.WriteLine("移动车厢: {0}从入轨到缓冲轨{1}.", c, bestTrack);  if(c < minH)  {  minH = c;  minS = bestTrack;  }  return true;  }  //若需出轨的编号恰是缓冲轨的最小编号, 则直接出轨.  static void Output(ref int minH, ref int minS, LinkStack<int>[] h)  {  h[minS].Pop();//从堆栈minS中删除编号最小的车厢minH  Console.WriteLine("移动车厢, {0}从缓冲轨{1}到出轨.", minH, minS);  //通过检查所有的栈顶,搜索新的minH和minS  minH = int.MaxValue;  minS = -1;  for(int i = 0; i < h.Length; i++)  {  if (h[i].IsEmpty() == false && h[i].StackTop < minH)  {  minH = h[i].StackTop;  minS = i;  }  }  }  //将入轨的编号放入缓冲轨.  for(int i = 0;i < p.Length; i++)  {  if (p[i] == nowOut)  {  Console.WriteLine("移动车厢: {0}从入轨到出轨.", p[i]);  nowOut++;  //从缓冲铁轨中输出  while(minH == nowOut)  {  Output(ref minH, ref minS, h);//出轨  nowOut++;  }  }  else  {  //将p[i]送入某个缓冲铁轨  if (Input(p[i], ref minH, ref minS, h) == false)  return false;  }  }  三、实验结果    四、结论与总结  结论:运用栈可以解决车辆重排的问题,本实验解决了如何将车厢编号打乱的列车通过缓冲轨道重排成新的有序列车.  总结:栈可以通过入栈操作将数据元素插入栈顶,出栈操作能够移除栈顶的数据元素,可以判断栈中是否包含数据元素, 可以获取栈中实际包含数据元素的个数, 可以移除栈中的所有数据元素,可以获取栈顶元素. |
|  |