1074. Reversing Linked List (25)

时间限制

400 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

Given a constant K and a singly linked list L, you are supposed to reverse the links of every K elements on L. For example, given L being 1→2→3→4→5→6, if K = 3, then you must output 3→2→1→6→5→4; if K = 4, you must output 4→3→2→1→5→6.

**Input Specification:**

Each input file contains one test case. For each case, the first line contains the address of the first node, a positive N (<= 105) which is the total number of nodes, and a positive K (<=N) which is the length of the sublist to be reversed. The address of a node is a 5-digit nonnegative integer, and NULL is represented by -1.

Then N lines follow, each describes a node in the format:

*Address Data Next*

where *Address* is the position of the node, *Data* is an integer, and *Next* is the position of the next node.

**Output Specification:**

For each case, output the resulting ordered linked list. Each node occupies a line, and is printed in the same format as in the input.

**Sample Input:**

00100 6 4

00000 4 99999

00100 1 12309

68237 6 -1

33218 3 00000

99999 5 68237

12309 2 33218

**Sample Output:**

00000 4 33218

33218 3 12309

12309 2 00100

00100 1 99999

99999 5 68237

68237 6 -1

[提交代码](https://www.patest.cn/contests/pat-a-practise/1074)

没什么好说的……记得把生成的链表末尾的next值重置为-1就可以了……因为可能存在经过倒置的链表结尾……

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#include<cstring>

#include<climits>

#include<cmath>

#include<fstream>

using namespace std;

#define MAX 100005

struct store

{

int data;

int next;

int pos;

};

store linklist[MAX];

vector<store>result;

int main()

{

int startpos, num, scope;

cin >> startpos >> num >> scope;

for (int i = 0; i < num; i++)

{

int pos, data, next;

scanf("%d %d %d", &pos, &data, &next);

linklist[pos].data = data;

linklist[pos].next = next;

linklist[pos].pos = pos;

}

int nowpos = startpos,cnt=0;

vector<store>temp;

while (1)

{

if (nowpos == -1)

break;

temp.push\_back(linklist[nowpos]);

nowpos = linklist[nowpos].next;

cnt++;

if (cnt == scope)

{

for (int i = 0; i < temp.size()/2; i++)

{

swap(temp[i], temp[temp.size() - 1 - i]);

}

for (int i = 0; i < temp.size(); i++)

{

result.push\_back(temp[i]);

}

temp.clear();

cnt = 0;

}

}

if (cnt)

{

for (int i = 0; i < temp.size(); i++)

{

result.push\_back(temp[i]);

}

}

for (int i = 0; i < result.size() - 1; i++)

{

result[i].next = result[i + 1].pos;

}

result.back().next = -1;

int i;

for (i = 0; i < result.size()-1; i++)

{

printf("%05d %d %05d\n", result[i].pos, result[i].data, result[i].next);

}

printf("%05d %d %d\n", result[i].pos, result[i].data, result[i].next);

}