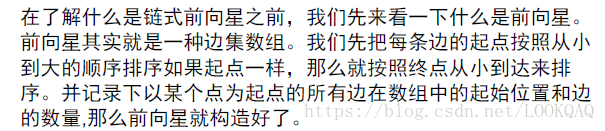
# 【链式前向星+存图】讲解

2018年07月31日 16:03:20 [Dreamers\_Boy](https://me.csdn.net/LOOKQAQ" \t "https://blog.csdn.net/lookqaq/article/details/_blank) 阅读数：3405

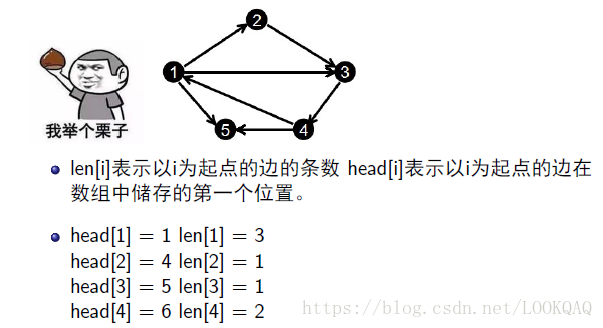
 版权声明：欢迎转载 ！只想安静地写代码圆梦！ https://blog.csdn.net/LOOKQAQ/article/details/81304637

大佬的链式前向星:[https://blog.csdn.net/acdreamers/article/details/16902023](https://blog.csdn.net/acdreamers/article/details/16902023" \t "https://blog.csdn.net/lookqaq/article/details/_blank)

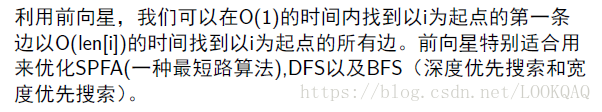
【前向星】：

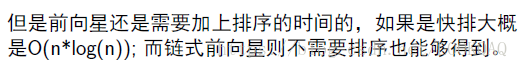


解释一下：



 【前向星和链式前向星的不同】：





 【给出链式前向星的代码实现】：

#include<bits/stdc++.h>

using namespace std;

#define MAXN 100501

struct NODE{

int w;

int e;

int next; //next[i]表示与第i条边同起点的上一条边的储存位置

}edge[MAXN];

int cnt;

int head[MAXN];

void add(int u,int v,int w){

edge[cnt].w=w;

edge[cnt].e=v; //edge[i]表示第i条边的终点

edge[cnt].next=head[u]; //head[i]表示以i为起点的最后一条边的储存位置

head[u]=cnt++;

}

int main(){

memset(head,0,sizeof(head));

cnt=1;

int n;

cin>>n;

int a,b,c;

while(n--){

cin>>a>>b>>c;

add(a,b,c);

}

int start;

cin>>start;

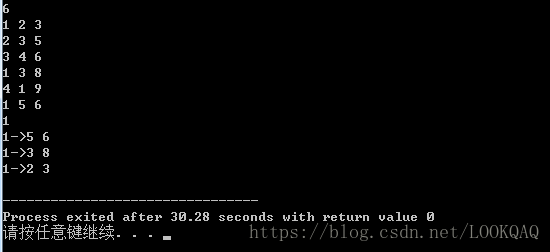
for(int i=head[start];i!=0;i=edge[i].next)

cout<<start<<"->"<<edge[i].e<<" "<<edge[i].w<<endl;

return 0;

}

 结果：



【分析一下】：

注意cnt的初值我们初始化为1 cnt

edge[1].next = head[1] = 0, head[1] = 1, 2 ;

edge[2].next = head[2] = 0, head[2] = 2, 3 ;

edge[3].next = head[3] = 0, head[3] = 3, 4 ;

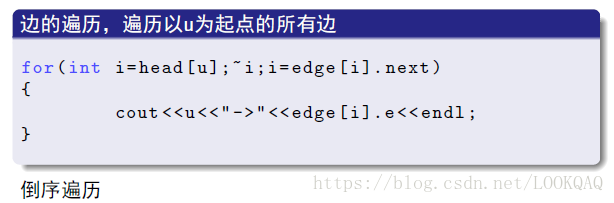
edge[4].next = head[1] = 1, head[1] = 4, 5 ;

edge[5].next = head[4] = 0, head[4] = 5, 6 ;

edge[6].next = head[1] = 4, head[1] = 6, 7 ;

****模拟：当start=1时，循环变为 for(i = head[1];i!= 0;i = edge[1].next) 通过输出语句将第六条边的终点和权值（就是边的值）输出出来，同时i = edge[6].next = 4(相当于一个链表将第四条边牵出来)，然后将第四条边的终点和权值输出，这是i = edge[4].next = head[1] = 1,又将第一条边牵出来，输出第一条边的终点和权值，然后i=0,循环结束，其他类似。****

 【边是倒序遍历的】：



贴个链接:[点击这里！](https://blog.csdn.net/acdreamers/article/details/16902023" \t "https://blog.csdn.net/lookqaq/article/details/_blank)（好理解的一篇博客）