

寻找链表中的最大值

生成一个M个节点的单链表，随机给每个节点赋值，依此寻找链表中的最大值，并把最大值的节点删除，并输出，然后再次寻找剩下的最大值，知道链表为空。如图
随机链表为 head->1->5->8->3->7->10->NULL

输出顺序为 10 8 7 5 3 1

```
typedef struct tagNODE{
    int data;
    struct tagNODE* next;
} STNODE;

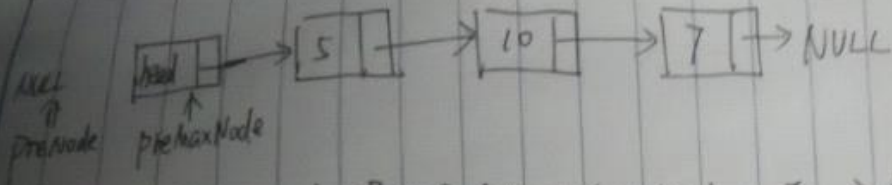
int main()
{
    int M=9;
    STNODE* Head;
    Head=(STNODE*)malloc(sizeof(STNODE));
    STNODE* Temp=Head;
    srand(time(0));
    for(int i=1;i<=M;i++) //尾插法建立单链表，随机赋值
    {
        STNODE* newNode=(STNODE*)malloc(sizeof(STNODE));
        newNode->data=rand()%100;
        Temp->next=newNode;
        newNode->next=NULL;
        Temp=newNode;
    }
    //打印原来的链表
    for(STNODE* temp=Head->next;temp!=NULL;temp=temp->next)
        printf("%d->", temp->data);
    printf("\n");

    while(Head->next!=NULL) //当所有节点都被删除那么结束
    {
        STNODE* preMaxNode; //存储最大值的前一个节点，用于删除最大值节点
        preMaxNode=Head; //默认最大值为第一个节点
        STNODE* preNode; //当前访问节点的前一个节点记录
        for(STNODE* temp=Head;temp!=NULL;preNode=temp, temp=temp->next) //从头
        几点开始循环访问
        { //每一次都记录前一个节点
            if(preMaxNode->next->data<=temp->data) //判断当前结点是否比已知最
            大值大
            {
                preMaxNode=preNode; //若是，则更新最大值节点的前一个节点
            }
        }
        printf("%d->", preMaxNode->next->data); //输出节点值

        //删除当前最大值节点
        STNODE* maxNode=preMaxNode->next;
        preMaxNode->next=maxNode->next;
        free(maxNode);
    }
}
```

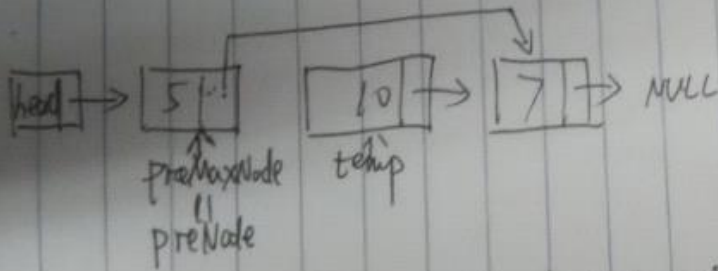
D:\visual_studio_Projects\OpencvOfmine\Opencv
22->13->30->10->49->16->91->37->41->
91->49->41->37->30->22->16->13->10->

求链表中的最大值并依次删除



用一个 preNode 指针来记录当前访问点的前一个 Node.

用一个 preMaxNode 指针保存当前最大值的前一个结点



10 为最大值, 那么 $preMaxNode = preNode$ 指向 5.

删除后, 重复操作

