## P3368 【模板】树状数组 2

### 1 原题目

题目描述 <https://www.luogu.com.cn/problem/P3368>

如题，已知一个数列，你需要进行下面两种操作：

将某区间每一个数数加上 x；

求出某一个数的值。

输入格式

第一行包含两个整数 N、M，分别表示该数列数字的个数和操作的总个数。

第二行包含 N个用空格分隔的整数，其中第 i 个数字表示数列第 i 项的初始值。

接下来 M 行每行包含 2 或 4个整数，表示一个操作，具体如下：

操作 1： 格式：1 x y k 含义：将区间 [x,y] 内每个数加上 k；

操作 2： 格式：2 x 含义：输出第 x个数的值。

输出格式

输出包含若干行整数，即为所有操作 2的结果。

输入输出样例

**输入 #1**

5 5

1 5 4 2 3

1 2 4 2

2 3

1 1 5 -1

1 3 5 7

2 4

**输出 #1**

6

10

说明/提示

样例 1 解释：



故输出结果为 6、10。

### 2 基础知识

#### 2.1数据结构

树状数组（解法一用到）

#### 2.2算法

无。

### 3 解法

#### 3.1 解法1（C++已过）

树状数组的区间更新，单点查询的模板运用。

参考链接：<https://www.cnblogs.com/xenny/p/9739600.html>

本人代码：

|  |
| --- |
| #include <iostream>  #include<algorithm>  #include <cstdio>  #include <cstdlib>  using namespace std;  int a[500005],c[500005],N;//对应原数组和树状数组  int lowbit(int x)  {      return x&(-x);  }  int getsum(int i)//求a[i]的值  {      int res = 0;      while(i > 0)      {          res += c[i];          i -= lowbit(i);      }      return res;  }  void updata(int i,int k)//在i位置加上k（此函数只更改树状数组的值，若要更改原数组的值需要在主函数中实现）  {      while(i <= N)      {          c[i] += k;          i += lowbit(i);      }  }  int main()  {      int M,i;      scanf("%d %d",&N,&M);      for(i = 1; i <= N; i++)      {          scanf("%d",&a[i]);          updata(i,a[i] - a[i - 1]);//输入初值的时候，也相当于更新了值      }      while(M--)      {          int o,x,y,k;          scanf("%d",&o);          if(o == 1)          {              scanf("%d %d %d",&x,&y,&k);              updata(x,k);//a[x] - a[x-1]增加k              updata(y + 1,-k);//a[y+1] - a[y]减少k          }          if(o == 2)          {              scanf("%d",&x);              printf("%d\n",getsum(x));          }      }      return 0;  } |

## P3805 【模板】manacher算法

### 1 原题目

题目描述

给出一个只由小写英文字符a,b,c,…y,z 组成的字符串 S ,求 S中最长回文串的长度 。

字符串长度为 n。

输入格式

一行小写英文字符 a,b,c,…y,z 组成的字符串 S。

输出格式

一个整数表示答案。

输入输出样例

**输入 #1**

aaa

**输出 #1**

3

说明/提示

1≤*n*≤1.1×10^7。

### 2 基础知识

Manacher马拉车算法（解法一用到）

### 3 解法

#### 3.1 解法1（C++已过）

马拉车算法的模板题目。

关于此算法参考参考以下两个链接：

1. <https://blog.csdn.net/Form_/article/details/79766986>

2. <https://www.cnblogs.com/czsharecode/p/9705358.html>

本人代码：

|  |
| --- |
| #include <iostream>  #include<algorithm>  #include <cstring>  #include <cstdio>  #include <cstdlib>  #include <cmath>  #include <vector>  using namespace std;  int Manacher(string &s)  {      //开头加'$'是为了防止越界      string t = "$#";      //在原字符串开头结尾以及字符之间插入一个字符(未在串中出现过的)      for(int i = 0; i < s.length(); i++)      {          t += s[i];          t += '#';      }      //p[i] - 1就是t串中以i为中点的回文长度      vector<int> p(t.length(),0);      //mx(某回文串延伸到的最右边下标),id(mx所属回文串中心下标)      int mx = 0,id = 0;      //bit（结果最大回文串中心下标）,maxlen（最大回文长度）      int maxlen = 0;      for(int i = 1; i < t.length(); i++)      {          //算法核心          p[i] = mx > i ? min(p[2 \* id - i],mx - i) : 1;          //一个个进行比较          //t数组最左边 = '$',最右边 = '\0'，无需判断边界          while(t[i + p[i]] == t[i - p[i]])              p[i]++;          //当t[i]匹配的 右边界超过mx时,mx和id就更新          if(i + p[i] > mx)          {              mx = i + p[i];              id = i;          }          //更新结果数据          if(p[i] > maxlen)          {              maxlen = p[i];          }      }      return maxlen - 1;  }  int main()  {      string s;      while(cin>>s)      {          printf("%d\n",Manacher(s));      }      return 0;  } |

### 1 原题目

### 2 基础知识

### 3 解法

#### 3.1 解法1（C++已过）

本人代码：

|  |
| --- |
|  |

### 1 原题目

### 2 基础知识

### 3 解法

#### 3.1 解法1（C++已过）

本人代码：

|  |
| --- |
|  |

### 1 原题目

### 2 基础知识

### 3 解法

#### 3.1 解法1（C++已过）

本人代码：

|  |
| --- |
|  |

### 1 原题目

### 2 基础知识

### 3 解法

#### 3.1 解法1（C++已过）

本人代码：

|  |
| --- |
|  |