

## 题目：高级数组实现

### 目录

1 高级数组-----	2
1.1 输入输出样例 -----	3
1.2 数据规模、内存限制与时长限制 -----	4
1.3 提交要求与考核标准 -----	4

## 1 高级数组

数组（长度为  $n$ ,  $A[1] \cdots A[n]$ ）是一个常见的编程数据结构，它可以实现某个地址上的值的修改。它需要  $O(n)$  的时间计算一个区间  $[i, j]$  内的和（ $A[i] + A[i + 1] + \cdots + A[j]$ ）。需要  $O(n)$  的时间修改一个区间的  $[i, j]$  内每一个值，使其每个数都加上一个给定的数  $C$ 。此外，需要  $O(n)$  的时间插入或者删除一个数。当然区间求和与区间求最值的代价基本相同，这里忽略。一个数组需要每一次操作都留下一个副本，才能保证有机会访问历史版本的数组内容。

现在请你实现一个高级的数组，它需要有如下功能，且每个操作的时间复杂度或均摊复杂度的期望不超过  $O(\log n)$ （注意：有 10% 的测试数据可以通过仅采用  $O(n)$  的算法来完成，详见 1.3.2 评分标准）：

- $sum(i, j)$  区间求和操作（ $sum$  简写为  $s$ ）  
输入  $i, j$ ，计算  $A[i] + A[i + 1] + \cdots + A[j]$
- $add(i, j, C)$  区间内每个数都增加  $C$ （区间修改操作），其中  $C$  为输入的整数（ $add$  简写为  $a$ ）  
输入  $i, j, C$ ,  $A[i], A[i + 1], \cdots, A[j]$  每个数都加上  $C$
- $insert(i, C)$  插入一个数到特定位置（ $insert$  简写为  $i$ ）  
输入  $i, C$ ，在  $A[i]$  位置插入  $C$ ，原来数字依次往后移动，最终数组总长度加一
- $delete(i)$  删除一个特定位置的数（ $delete$  简写为  $d$ ）  
输入  $i$ ，删除  $A[i]$ ，原来数字依次往前移动，最终数组总长度减一
- $checkout(t)$  切换副本（ $checkout$  简写为  $c$ ）  
输入  $t$ ，把当前数组切换到第  $t$  次操作后的副本（ $t$  为 0 时为初始化状态，以后每次操作都会产生副本，包括  $sum$  和  $checkout$  操作）。

注意：请确保你的代码是在线算法，即在处理完上次操作后再读取下个操作。

本题请写一个数据结构结构体或类 `SuperArray`，并实现它的  $sum(i, j)$ ,  $add(i, j, C)$ ,  $insert(i, c)$ ,  $delete(i)$ ,  $checkout(t)$  等功能。请写一个程序来调用你自己的结构体或类，从标准输入读取操作指令，并以此来操作你的结构体或类（针对部分语言中保留字或关键字，可以根据情况修改函数名称，例如 `c++` 中用 `erase` 代替 `delete`）。

若代码没有实现结构体/类和其中的函数，则不得分。若代码没有结构体/类的抽象，则不得分。测试数据为合法的操作和数据。程序可以针对异常处理（例如访问越界、非法操作类型等）可以自行定义处理方法。

## 1.1 输入输出样例

输入第一行为  $T$ ，表示一共有  $T$  组测试。

每组测试的第一行为  $n, m$ ，表示原数组长度和操作数目，数组初始值为 0。

每组测试后续有  $m$  行，每行表示一个操作，用操作**首字母**表示操作类型，并在后面附上操作的参数，例如  $s\ 2\ 3$  表示  $sum(2, 3)$ 。

每个  $sum(i, j)$  对应一行输出，每行都是一个整数，表示查询区间和的操作得到的答案。

**样例-1:**

Sample Input	Sample Output
2	6
1000 6	4
a 1 4 2	0
s 1 3	16
c 1	15
s 3 4	
c 0	
s 1 4	
5 9	
a 1 1 1	
a 2 2 2	
a 3 3 3	
a 4 4 4	
a 5 5 5	
i 5 6	
s 1 5	
d 1	
s 1 4	

**样例说明:**

有 2 组测试。第一组测试数组初始长度为 1000，6 个操作，第一个测试主要测切换副本功能。第二个测试数组初始长度为 5，主要是测试插入删除功能。

## 样例-2:

Sample Input	Sample Output
1 100000 10 a 1000 2000 -1000 d 1500 i 1200 3000 s 1 2000 c 0 d 1800 c 6 d 1600 a 1000 2000 2000 s 1 2000	-997000 2002000

## 样例说明:

有 1 组测试。测试数组比较大的时候，各种操作之后的输出结果。

## 1.2 数据规模、内存限制与时长限制

对于所有数据满足所有输入均为整数， $1 \leq T \leq 10$ ， $1 \leq n \leq 10^5$ ， $1 \leq m \leq 10^5$ 。

- 对于 10% 的数据， $1 \leq m \leq 10000$ ， $1 \leq n \leq 10000$ 。
- 对于 10% 的数据，不涉及插入删除或者切换副本操作。
- 对于 10% 的数据，不涉及区间修改或切换副本操作。
- 对于 10% 的数据，不涉及插入删除操作。
- 对于剩下 60% 的测试数据，所有操作均需要支持，且  $-2^{40} \leq C \leq 2^{40}$ 。

计算结果不超过 `int64` 的范围，请放心使用 `int64`。

请确保你的程序在 2 秒内运行一组测试并输出正确的结果。

你的程序使用内存原则不超过 1 GB。

## 1.3 提交要求与考核标准

### 1.3.1 代码要求

推荐 C/C++、Java、Rust 等，不限定编程语言和开发环境，但请确保你使用的语言或代码具有足够高的运行效率（除样例外，你可以自行构造测试例进行测试）。提交的是源代码文件，输入直接来自标准输入 `STDIN`，输出到标准输出 `STDOUT` (`./your_program < sample.in > sample.out`)。第三方库不被允许使用，涉及系统的操作除了输入输出外不被允许使用，不允许多线程或者多进程，注意你的程序没有 `root` 权限。

请把你的程序命名为 `superarray.xxx`，其中 `xxx` 后缀名请根据你的语言选择，文件名请全部小写，例如 `superarray.cpp`。允许在合理范围调用自己实现的模板类头文件（例如 `h` 文件）或者自己实现的库，但是提交的库文件等会被代码查重。

请注意评测机仅仅安装了官方标准库，且禁止安装第三方库，评测机不联网且采用单 CPU 测试，没有 GPU。

### 1.3.2 评分标准

设计和实现程序，完成前述功能。同时请注意代码规范、命名标准、注释和说明。

评测分为程序正确和性能分（90 分）和文档和程序规范分（10 分）两部分，合计 100 分。

程序正确和性能分由机器打分，一共十个输入文件，对于每一个输入文件，如果能在给定时间给定资源的前提下，成功算出正确结果并输出，即可得到对应输入文件的正确和性能分数，否则对应输入文件得分为零。

- 对于 10% 的数据， $1 \leq m \leq 10000$ ， $1 \leq n \leq 10000$ 。（可以仅采用普通  $O(n)$  算法来完成，通过测试可得 20 分）。
- 对于 10% 的数据，有求和与区间修改，不涉及插入删除或者切换副本操作。（采用  $O(\log n)$  算法来完成，通过测试可得 20 分；如果仍采用  $O(n)$  算法，可能因超时而无法通过测试，下同）。
- 对于 10% 的数据，不涉及区间修改或切换副本操作，但会涉及插入或删除与求和操作（采用  $O(\log n)$  算法来完成，例如采用平衡树支持插入或删除，通过测试可得 10 分）。
- 对于 10% 的数据，不涉及插入删除操作，但会涉及求和、区间修改或切换副本操作（采用  $O(\log n)$  算法来完成，通过测试可得 10 分）。
- 对于剩下 60% 的测试数据，所有操作均需要支持，且  $-2^{40} \leq C \leq 2^{40}$ 。（均采用  $O(\log n)$  算法来完成，通过测试可得 30 分）。

如果不能完成全部程序功能，也请不要担心，我们会根据你完成的测试点酌情给分。如果程序有错误，也可能酌情给分。

文档和程序规范分（10 分）：除了注意源代码的代码规范、命名标准、注释之外，请你在程序源代码文件开始部分采用注释的方式增加一段说明文本（中英文均可），**简要阐述你的源代码编译过程，以及你采用高级数组操作的主要设计思路及复杂度分析。**

### 1.3.3 提交要求

请将程序源代码文件使用 7z 格式压缩后命名为“姓名.7z”并执行以下两个操作：

- 将其放置在 U 盘中，请确保文件完整性。

压缩包中应仅包含完整的源代码文件，不包含测试用的输入输出、中间文件或可执行文件。压缩包大小原则上不超过 5MB。