Solution 6

1、动态数组是一种支持平摊$O(1)$插入和删除的数据结构。插入时，数组大小如果超过数组容量，则额外申请2倍的空间，并把数据复制过去；删除时，如果数组大小$\leq$容量的$\frac{1}{4}$，则将数组容量缩小为原来的$\frac{1}{2}$，并把数据复制过去。使用势能分析法说明删除操作的复杂度也是平摊$O(1)$。

定义势能函数 $\Phi(n,c)=\begin{cases}2n-c & \text{if } n\geq\frac{c}{2}\\ c-2n & \text{if } n<\frac{c}{2}\end{cases}$

对于`grow`操作 $ac_i=c_i+\Delta\Phi(n,c)=c+(2n-2c-2n+c)=0$

对于`shrink`操作 $ac_i=c_i+\Delta\Phi(n,c)=\frac{c}{4}+(2n-\frac{c}{2}-c+2n)=4n-\frac{5}{4}c=-\frac{1}{4}c$

对于`append`操作 $ac_i=1+2=3$

对于`erase`操作 $ac_i=1+2=3$

在$n=\frac{c}{2}$时势能最小，n增加和减少分别在为生长和收缩积累势能。

2、**2-3树**的定义是：

如果一个内部节点拥有一个数据元素、两个子节点，则此节点为**2节点**。

如果一个内部节点拥有两个数据元素、三个子节点，则此节点为**3节点**。

当且仅当以下叙述中有一条成立时，*T*为2–3树：

- *T*为空。即*T*不包含任何节点。
- *T*为拥有数据元素*a*的2节点。若*T*的左子节点为*L*、右子节点为*R*，则：
  - *L*和*R*是等高的2–3树；
  - *a*大于*L*中的所有数据元素；
  - *a*小于等于*R*中的所有数据元素。
- *T*为拥有数据元素*a*和*b*的3节点，其中*a* < *b*。若*T*的左子节点为*L*、中子节点为*M*、右子节点为*R*，则：
  - *L*、*M*、和*R*是等高的2–3树；
  - *a*大于*L*中的所有数据元素，并且小于等于*M*中的所有数据元素
  - *b*大于*M*中的所有数据元素，并且小于等于*R*中的所有数据元素。

2-3树支持查找`find`、插入`insert`和删除`delete`三种操作，时间复杂度均为$O(\log n)$，如果不清楚具体实现可以自行上网查询。使用势能分析法说明插入和删除操作的时间复杂度都是平摊$O(\log n)$。

因为树的高度是$O(\log n)$，所以插入和删除都是$O(\log n)$。（插入和删除不用平摊直接就是$O(\log n)$，只要给出合理的证明都算对，不用平摊分析）

2-3中有一个节点分裂`split`的操作，`split`的次数也是均摊$O(1)$的，下面给出证明：

定义势能函数 $\Phi=3\text{结点的数量}$。因为每次操作最对加1个子节点，每次分裂都会使一个3节点分裂成2个2节点，所以$\Delta\Phi\leq 1-\text{分裂次数}$。
$\text{amortized }\#\text{splits}=\#\text{splits}+\Delta\Phi\leq\#\text{splits}+(1-\#\text{splits})=1$

对于**B树**，定义$\Phi=\text{有B个子结点的结点的数量}$

对于**(a,b)-树**，定义$\Phi=\text{有b个子结点的结点的数量}$

3、设计一个队列，支持入队`enqueue`，出队`dequeue`和查询最小值`find_min`，每个操作都是平摊$O(1)$，并用势能分析法分析复杂度。

双端队列`deque`可以使用类似动态数组的方式维护。对于一个队列同时维护头指针和尾指针，则可以同时支持头部插入删除，尾部插入删除，使用循环队列的方式可以节省空间，并且保证长度为n的数组一定可以维护长度$\leq n$的`deque`。当尾指针追上头指针时（即队列长度超过数组容量），则额外申请一倍的空间，删除类似。插入和删除的复杂度分析和动态数组的复杂度分析相同。

注意到全局最小值等价于最长的后缀的最小值。另开一个双端队列维护所有的后缀最小值。当插入一个元素时，从队列尾部弹出元素直到队列为空或者队列的尾部>插入的元素。当弹出一个元素时，检查队头的元素是否为弹出的元素，是则弹出，否则不做任何事。查询最小值相当于查询队头元素，复杂度$O(1)$。

4、设计一个数据结构，动态维护一个大小为n的集合S，支持插入一个元素`insert`，和删除最小的$\lceil \frac{n}{2} \rceil$个元素`remove_bottom_half`，每个操作都是平摊$O(1)$，并用势能分析法分析复杂度。

假设元素各不相同。

定义势能函数 $\Phi(S) = (2a+1)|S|$，其中a为找中位数的常数。

插入操作，直接插入 $ac_i = 1 + 2a + 1 = 2a + 2$

删除操作，用$O(n)$找中位数，把大于中位数的数复制到另一个数组，$c_i = an + \lfloor \frac{n}{2} \rfloor$，$ac_i = an + \lfloor \frac{n}{2} \rfloor - \lceil \frac{n}{2} \rceil (2a+1) \leq 0$

（考虑插入相同元素可以用哈希表去重，这里不做要求）

5、瑟尼欧里斯是将特殊护身符按特定顺序连接起来制成的。

经过 500 多年的时间，圣剑现在状况不佳，因此威廉决定彻底检查它。

瑟尼欧里斯有n个护身符。威廉将它们排成一行，其中第i个是整数a_i，初始均为0。

为了维护它，威廉需要执行m次`assign(l, r, x)`操作。每次对于所有$l \leq i \leq r$，将a_i赋值为x，每次操作的消耗定义为区间$[l, r]$里不同a_i的个数。

(a) 使用势能分析法说明m次操作的消耗最多是$3m$。

把值相同的区间看做一个结点，则任意区间$[l, r]$可以拆分成若干个结点，结点对应的区间互不相交。显然对于任意$[l, r]$结点的数量一定$<$ $[l, r]$里不同a_i的数量。

定义势能函数 $\Phi(S) = S$状态下，$[1, n]$内结点的数量。

对于`assign`操作，会将$[l, r]$变成一个结点，消耗$c_i =$原本$[l, r]$内结点的数量（记为k），插入后会最多增加3个结点（本身一个，加上切断原本两端最多2个区间）。$ac_i \leq k - (|S| - |S| + k - 3) = 3$。

(b) 说明任意小于3的常数都不成立。

考虑这样一种构造：

首先将$[\lfloor \frac{n}{2} \rfloor, n]$染为1，再将$\lfloor \frac{n}{2} \rfloor$这个点染为2，之后第$i$次操作选择$[\lfloor \frac{n}{2} \rfloor - i, \lfloor \frac{n}{2} \rfloor + i]$染为$i+2$，则之后的每次操作代价都是3，均摊$ac_i = \frac{1+1+3x}{x+2} (x = \lceil \frac{n}{2} \rceil - 2)$，$\lim_{n \to \infty} ac_i = 3$，因此对于任意$c < 3$，总是可以构造一种染色方法使得$c$不成立。

扩展阅读：**珂朵莉树（Chtholly Tree）**https://oi-wiki.org/misc/odt/