The background is a soft, painterly illustration of a young woman with long, straight blue hair and large, expressive brown eyes. She is wearing a white long-sleeved shirt under a purple garment, with a thick yellow scarf wrapped around her neck. Her hands are clasped together in front of her, holding a small, dark green leafy plant. The setting is a snowy landscape with bare, snow-laden tree branches visible on the left. The overall color palette is cool, dominated by blues, whites, and purples, with the yellow scarf providing a warm contrast. The text is centered over the image.

数据结构

复旦大学白学研究会

温昕岳

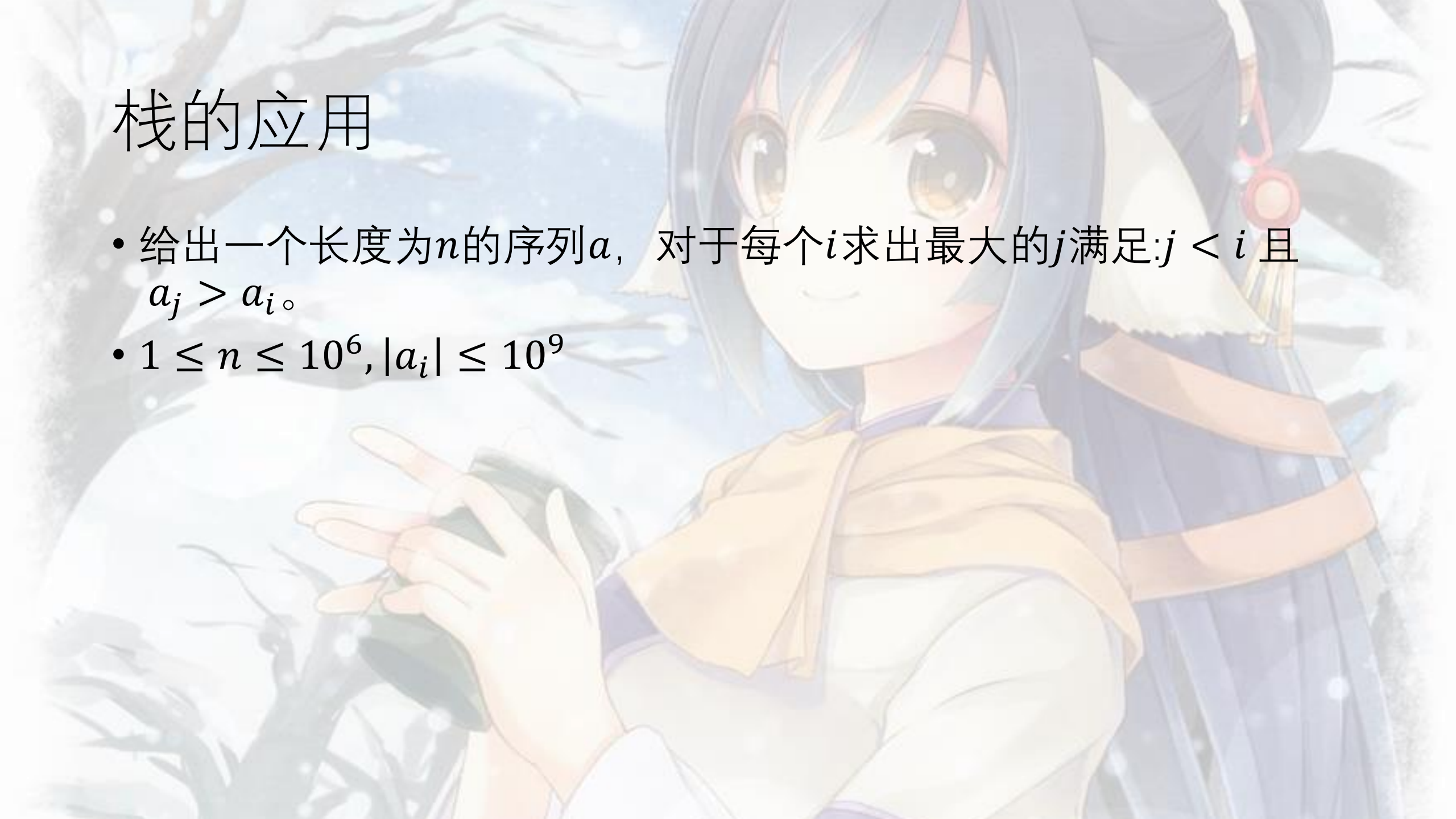
基本数据结构

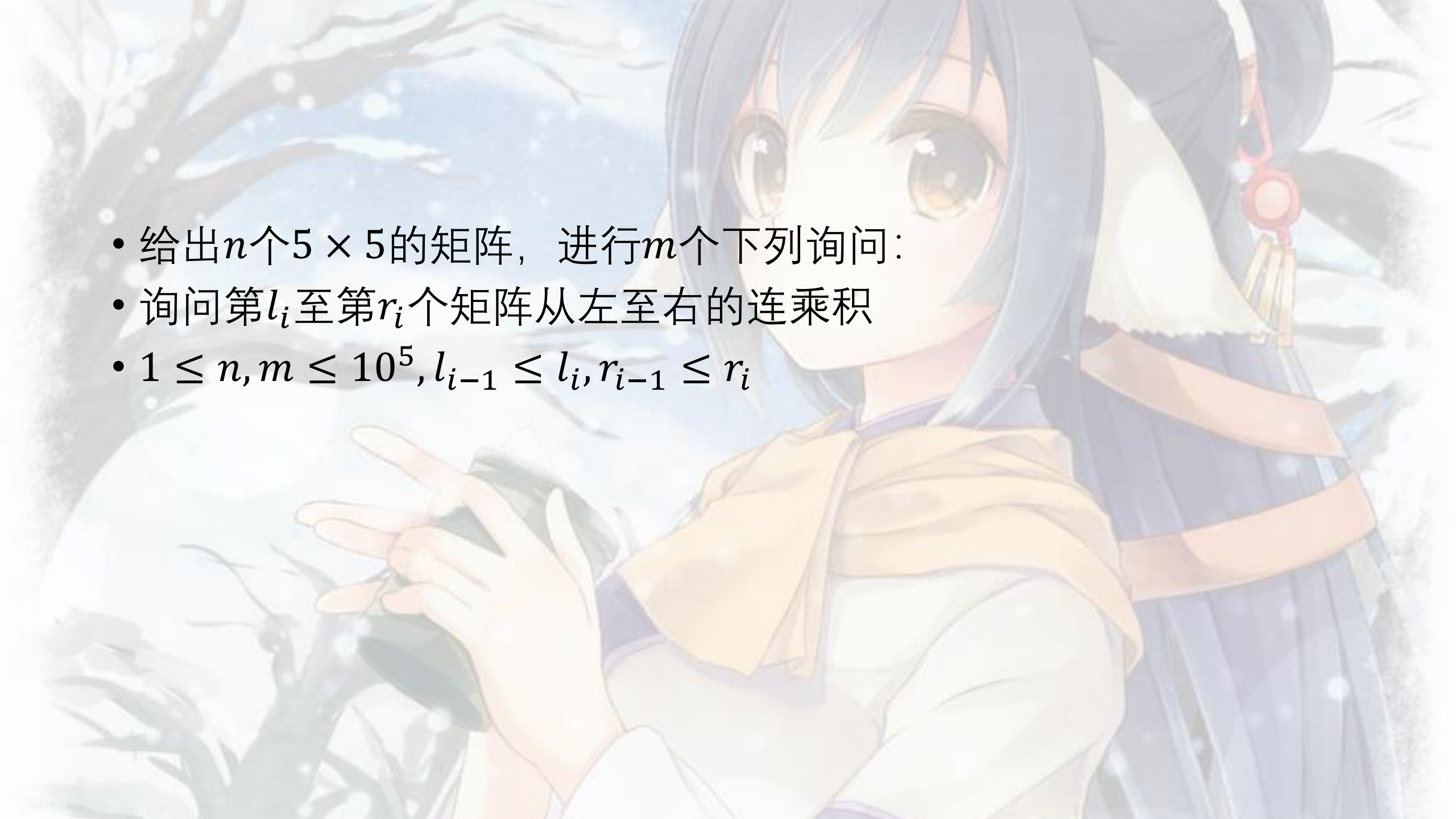
- 队列：元素满足先进先出原则
- 栈：元素满足先进后出原则



栈的应用

- 给出一个长度为 n 的序列 a , 对于每个 i 求出最大的 j 满足: $j < i$ 且 $a_j > a_i$ 。
- $1 \leq n \leq 10^6, |a_i| \leq 10^9$



- 
- 给出 n 个 5×5 的矩阵，进行 m 个下列询问：
 - 询问第 l_i 至第 r_i 个矩阵从左至右的连乘积
 - $1 \leq n, m \leq 10^5, l_{i-1} \leq l_i, r_{i-1} \leq r_i$

栈模拟队列

- 维护栈A和栈B两个栈
- push: 栈B push
- pop: 栈A pop, 若栈A为空, 将栈B中的所有元素push进栈A, 清空栈B。
- 栈中每个元素维护从自身到栈底的所有元素的连乘积。
- 查询时合并两个栈对应的矩阵
- 时间复杂度 $O((n + m)5^3)$ 。

二叉堆

- 二叉堆是一个满足特殊性质的二叉树
- 堆的性质：
 - 根节点的权值大于等于子树中任一节点的权值
 - 除了最底层之外，每层都被填满。即二叉堆是一棵完全二叉树。
- 堆的操作：
 - 获取最大值
 - 新插入一个节点
 - 删除最大值

新建一个二叉堆

- 假设 $[x+1, n]$ 已经满足堆的性质
- 将 x 下沉以使 $[x, n]$ 满足堆的性质
- 高度为 i 的点至多下沉 $i - 1$ 次
- $\sum_{i=1}^{\log_2 n} \lfloor \frac{n}{2^{i-1}} \rfloor (i - 1) \leq \sum_{i=1}^{\log_2 n} \frac{n}{2^{i-1}} (i - 1) \leq n \sum_{i=0}^{\log_2 n - 1} \frac{i}{2^i} < 2n$
- 时间复杂度 $O(n)$

分块

- 给出一个长度为 n 的数字序列，支持单点修改与前缀和查询。
- 将序列分成 B 块，每块的大小为 $O(\frac{n}{B})$
- 每块维护块中所有元素的和
- 修改时找到对应块 $O(1)$ 修改
- 查询时累加整块的和，最后一块暴力，时间复杂度 $O\left(B + \frac{n}{B}\right)$ 。
- 取 $B = \sqrt{n}$ ，查询有最优时间复杂度 $O(\sqrt{n})$

树状数组

- 第 i 块维护 $(i - \text{lowbit}(i), i]$ 的和
- 查询 $O(\log n)$
- 更新时, 找到所有的 i 满足 $x \in (i - \text{lowbit}(i), i]$
- $x += \text{lowbit}(x)$ 得到最小的 i 满足 $x \in (i - \text{lowbit}(i), i]$
- 修改 $O(\log n)$



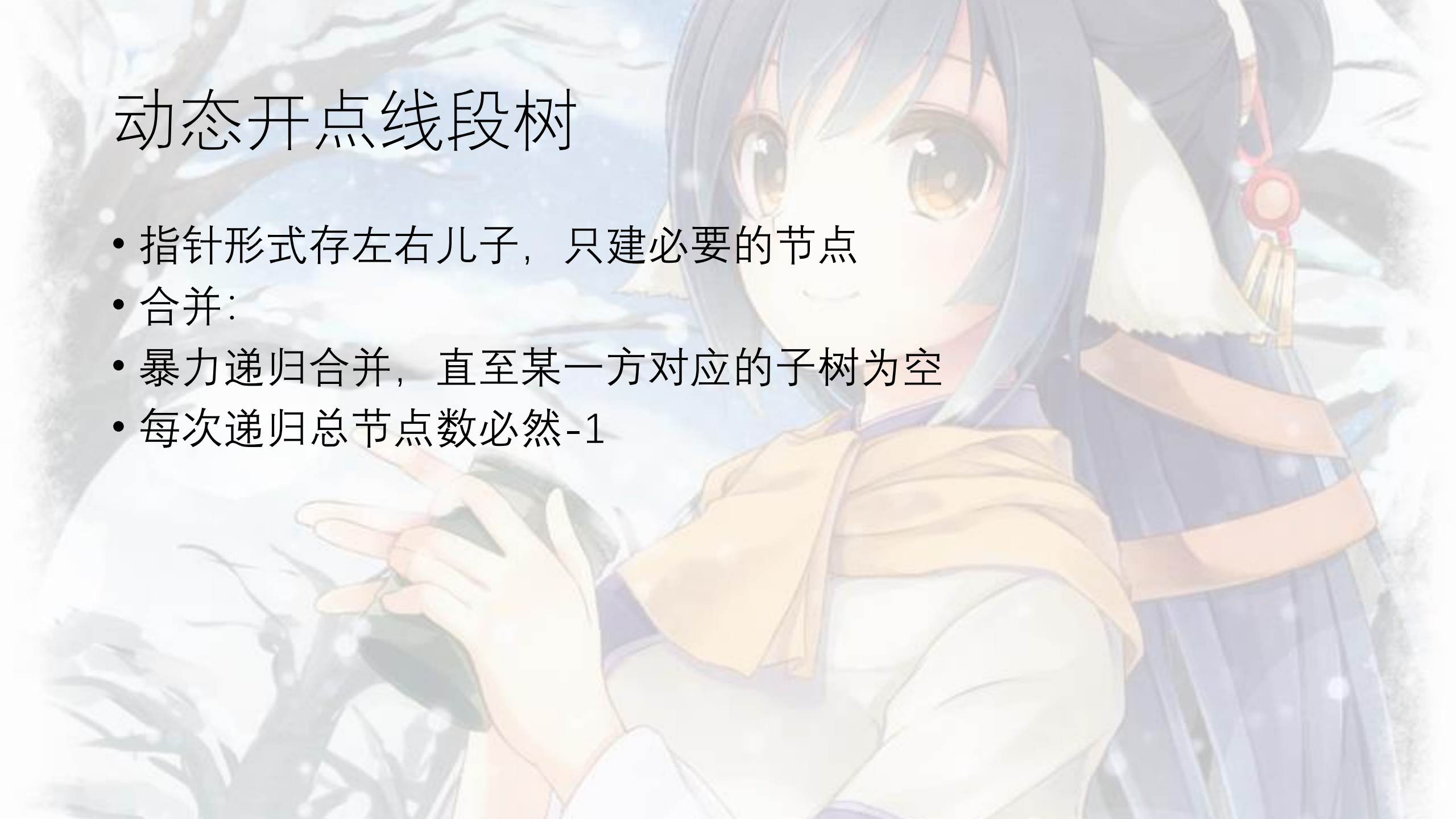
线段树

- 线段树是一种静态的平衡二叉树
- 快速区间修改：懒标记
- 标记处理方式：标记永久化/标记下传



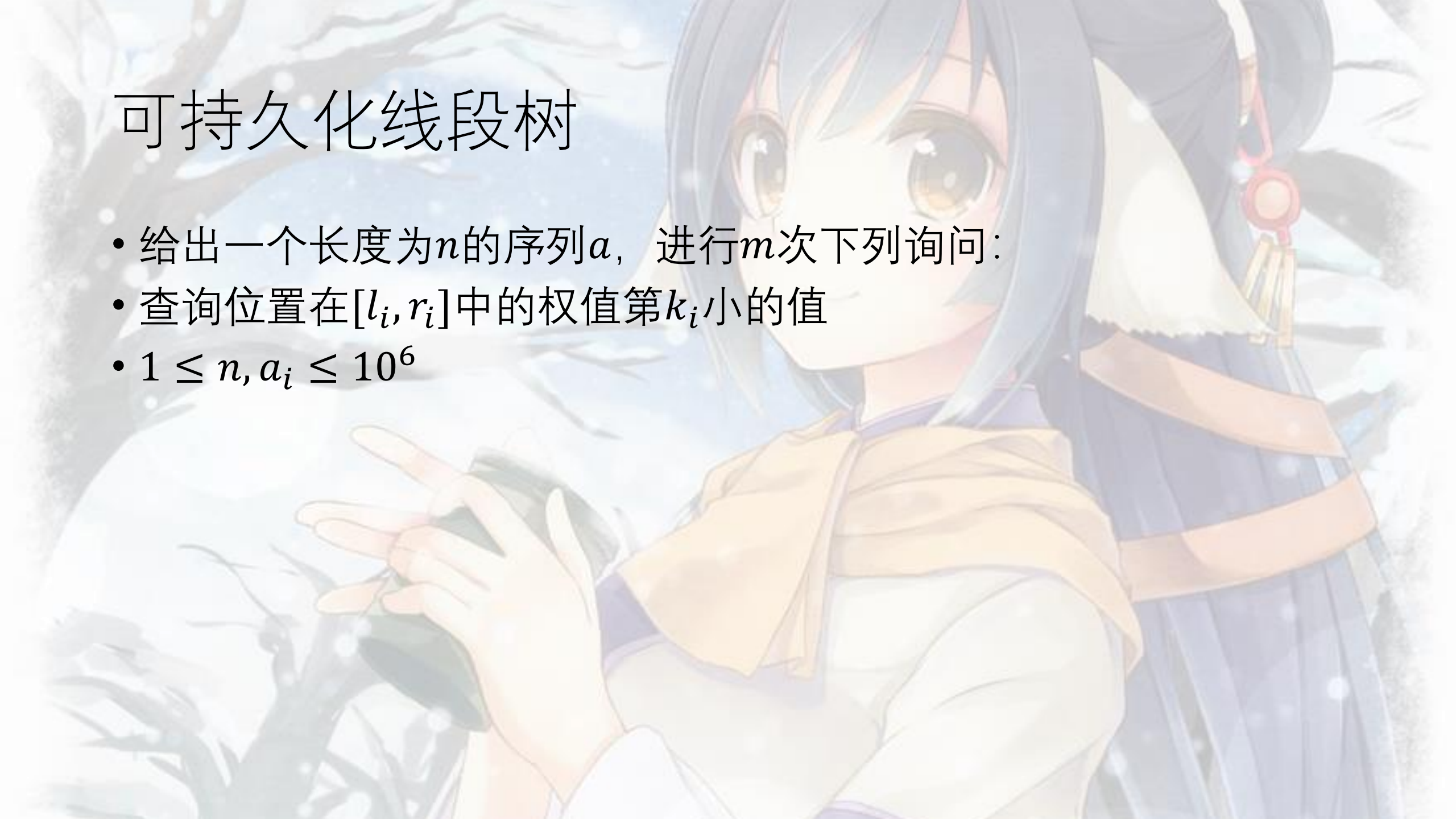
动态开点线段树

- 指针形式存左右儿子，只建必要的节点
- 合并：
- 暴力递归合并，直至某一方对应的子树为空
- 每次递归总节点数必然-1



可持久化线段树

- 给出一个长度为 n 的序列 a ，进行 m 次下列询问：
- 查询位置在 $[l_i, r_i]$ 中的权值第 k_i 小的值
- $1 \leq n, a_i \leq 10^6$



Treap

- 插入
- 删除
- 求第k大的元素
- 序列的合并与分割



Treap的合并

- 对Treap使用启发式合并，期望总时间复杂度为 $O(n \log^2 n)$
- 以两个根中权值更大的一个为新的根，把另一棵树按键值剖分为两棵树，递归到左右子树合并。期望总时间复杂度为 $O(n \log n)$

Splay Tree

- 插入
- 删除
- 求第k大的元素
- 序列的合并与分割

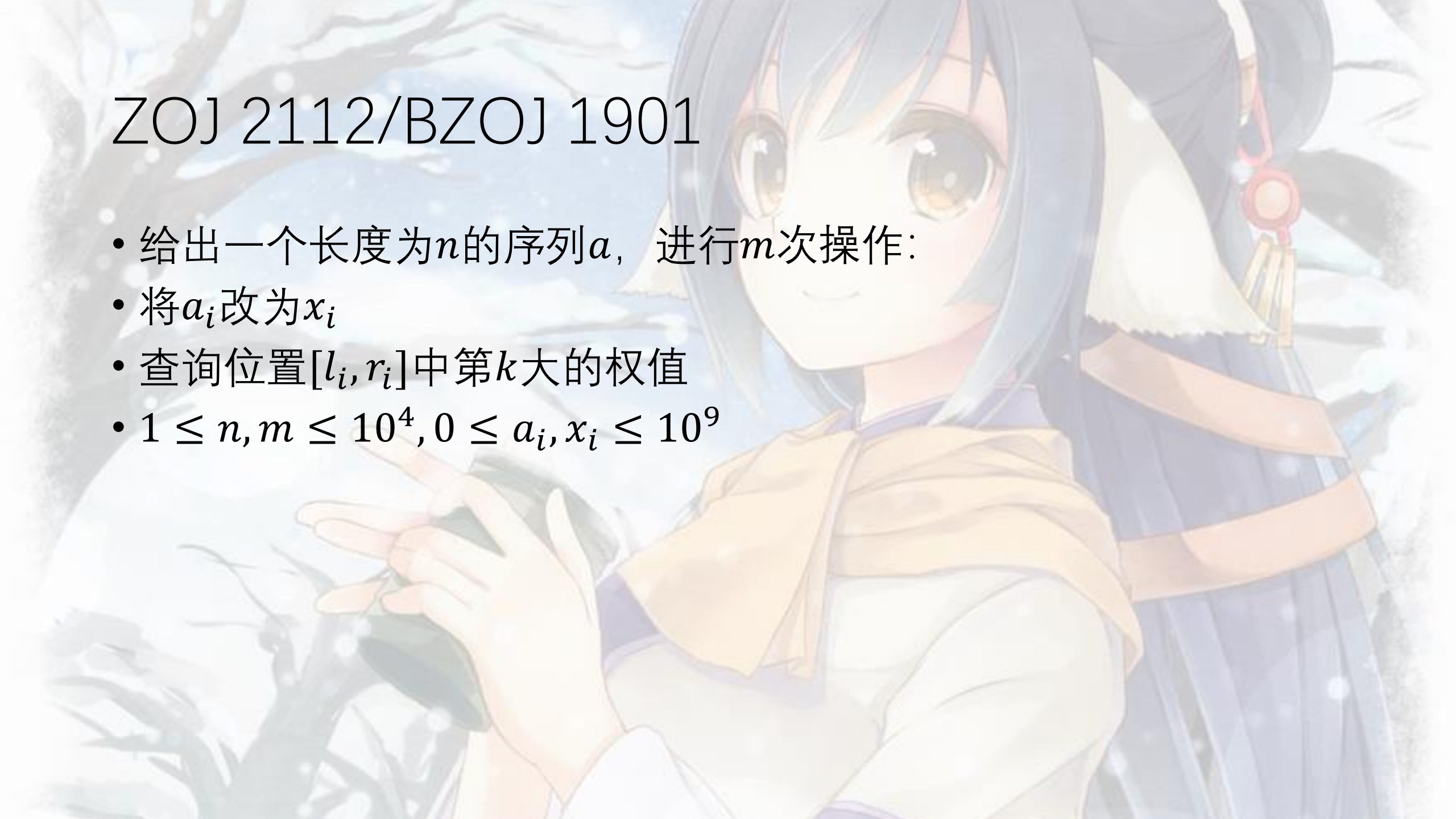


Splay Tree的合并

- 对Splay Tree使用启发式合并：对Splay Tree使用启发式合并的总时间复杂度为 $O(n \log^2 n)$ 。
- 如果将小的Splay Tree中的元素从小到大（即中序遍历）插入另一棵Splay Tree中，时间复杂度可以证明是 $O(n \log n)$ 的。具体证明和Dynamic Finger Theorem有关，在此仅作为一个有用结论给出。

ZOJ 2112/BZOJ 1901

- 给出一个长度为 n 的序列 a ，进行 m 次操作：
- 将 a_i 改为 x_i
- 查询位置 $[l_i, r_i]$ 中第 k 大的权值
- $1 \leq n, m \leq 10^4, 0 \leq a_i, x_i \leq 10^9$



嵌套数据结构

- 没有修改：线段树套vector
- 带修改：线段树套平衡树
- 另一种思路：树状数组套线段树



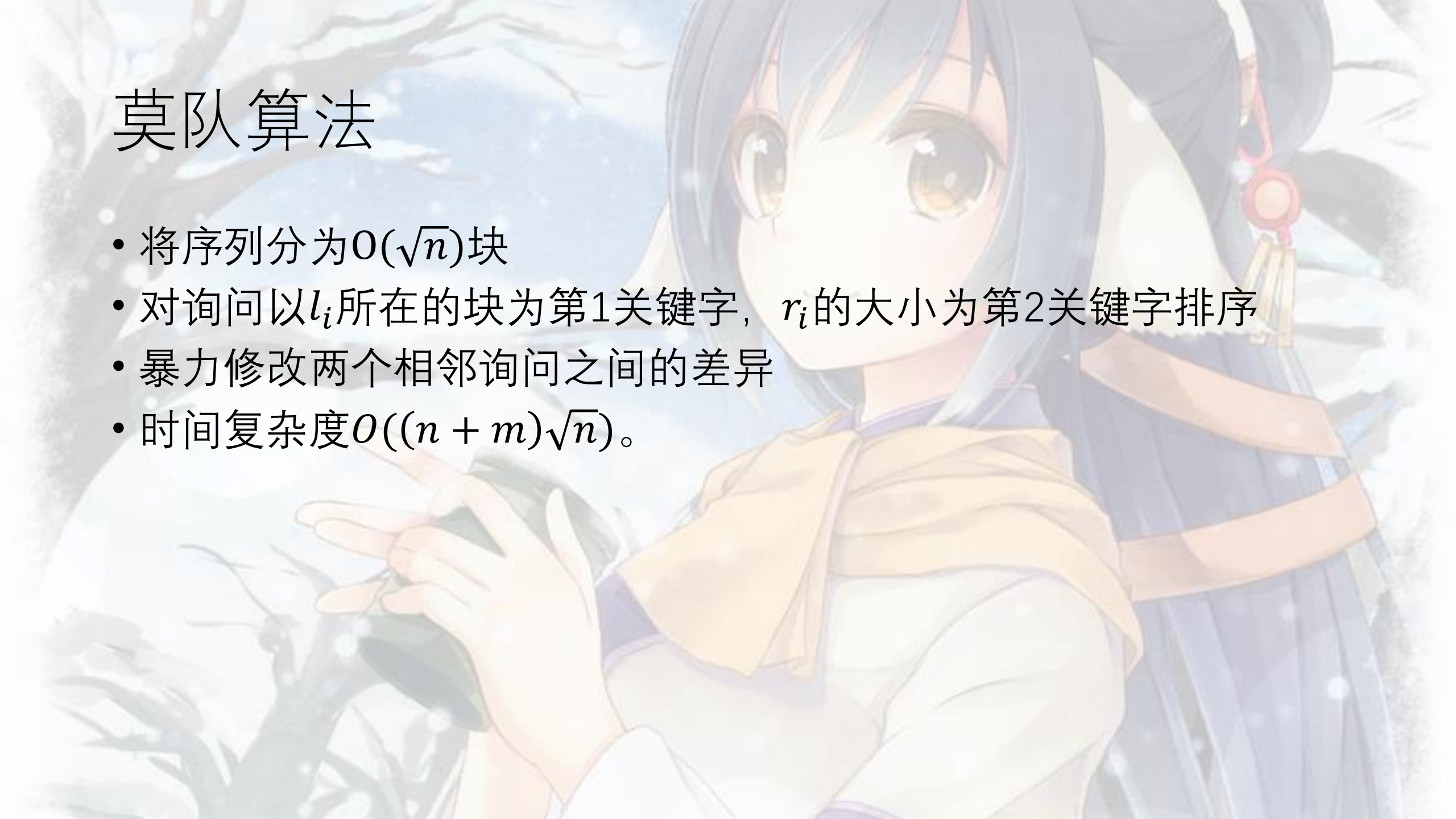
BZOJ 2038

- 给出一个长度为 n 的序列 c 和 m 个询问，每次询问从 $[l_i, r_i]$ 选两个位置，其权值相同的概率是多少。
- $1 \leq n, m \leq 50000, 1 \leq c_i \leq n$



莫队算法

- 将序列分为 $O(\sqrt{n})$ 块
- 对询问以 l_i 所在的块为第1关键字, r_i 的大小为第2关键字排序
- 暴力修改两个相邻询问之间的差异
- 时间复杂度 $O((n + m)\sqrt{n})$ 。



分治

- 将问题分割为若干个子问题，并对子问题的解做适当处理得到原问题的答案。
- 应用：
- 归并排序
- 求逆序对个数



BZOJ 1176

- 支持 q 次下列操作：
- 1. 在 (x, y) 处 $+w$
- 2. 查询以 (x_1, y_1) 为左下角， (x_2, y_2) 为右上角的矩形内的权值和。
- $1 \leq q \leq 10^5$

