Project-Shopping-Cart

冯睿骐
学号: 20307110161

实验环境

- **编译器**：gcc version 12.2.0 (Rev1, Built by MSYS2 project)
- **系统信息**
  - 版本 Windows 11 家庭中文版
  - 版本 21H2
  - 操作系统版本 22000.1219
- **机器信息**
  - 设备名称 yoga14s
  - 处理器 AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz
  - 机带 RAM 16.0 GB (15.4 GB 可用)
  - 系统类型 64 位操作系统, 基于 x64 的处理器

使用的数据结构

树用于存储购物车信息

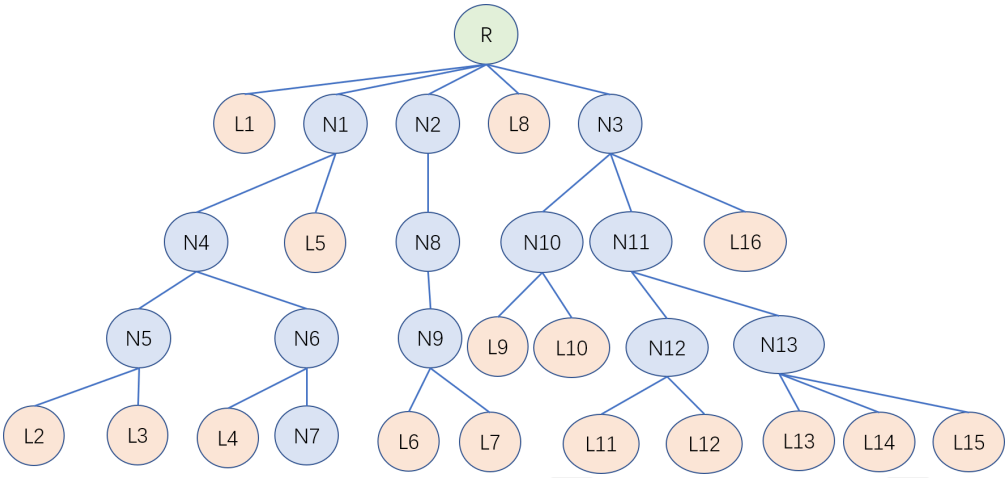
- 因为我允许商城内存在*多个不同优先级*的跨店满减和跨店折扣，进行优先级排序的过程在逻辑上和树十分类似，所以选择使用树来实现购物车
- 树的叶子节点用于储存实际在购物车内的商品，中间节点用于指示这个子树的叶子共有的优先级
- 节点结构体定义如下

```
struct TreeNode{
  TreeNode* sibling;  // 用左子女右兄弟的方式存储这颗树
  TreeNode* son;      //
  TreeNode* parent;   // 因为有较多取父母操作，维护一个指向父母节点的指针
  Item* item;         // 指向商品的指针；
  int data;           // 这个节点的最高优先级，下文将详细解释；叶子节点的data为-1
  int number;         // 指的是这件商品的数量；非叶子节点的number为0
  int add_time;       // 对叶子节点，指添加时间；对其他节点，指这个节点下所有叶子节点中最新的添加时间
  ... // 省略了构造函数
};
```

- 为了满足优先级的要求，在增删操作中需要维护这颗树，使它满足如下规则

1. 某个中间节点下方的所有叶子都要有这个中间节点对应的*优先级信息*[1]，且越子节点的*优先级*一定小于父节点的优先级
2. 任何一个节点的各个孩子节点中，`add_time` 从左至右升序排列[2]
3. 对于任意一个叶子，对它的祖先节点中的优先级从大到小排序[3]得到的序列称为$\{x_i\}$。将它放到任意一个树中允许的其他位置，同样操作，得到优先级序列$\{y_i\}$。如果两个序列不一样长，应该把短的序列用足够大的值补上。要求$\forall i, (x_i \leq y_i$ 或 $\exists j < i, x_i < y_i)$。[4][5][6]
4. 若某个叶子节点没有sibling,它上方一定是根节点。

- 一个例子如下图所示



  - 节点用来表示某个优先级，比如，`N1` 的叶子同跨店折扣，`N2` 的叶子同商店，而 `N5` 的叶子同跨店满减
  - 同层的节点，比如 `N1`，`N2` 和 `N3` 可以有不同的优先级，只要满足上面的规则 `3` 即可。
  - 中间节点可以没有sibling，只要下方的某个节点有siblings即可，比如图中的 `N2`，`N8`，`N9`
  - 每个中间节点的 `data` 值是它对应的优先级
- 按照上述条件，最终从左至右遍历打印*叶子*节点即为展示购物车需要的顺序

数组用于存储折扣信息

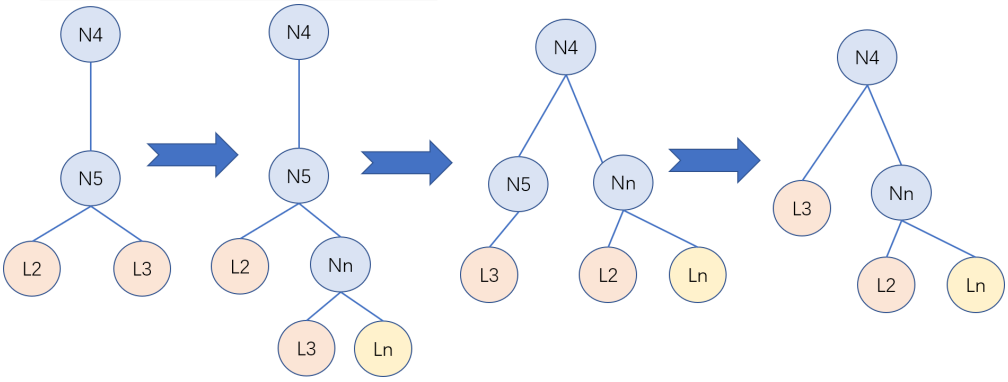
- 因为折扣信息需要的更改较不频繁，所以使用静态数组进行存储[7]
  - 具体而言，我使用了两个 `vector<vector<string>>` 来保存折扣和满减，其中内层的每一个 `vector<string>` 都存放着*相同优先级*的折扣/满减信息
- 商品可以通过一个数组对折扣信息进行索引，便于不同商品之间折扣信息的比较

算法逻辑

添加商品

- 假设不同优先级的优惠数量远少于购物车中商品数量，则
  - 平均时间复杂度: $O(N)$, 其中N是购物车中商品件数
  - 平均空间复杂度: $O(N)$, 其中N是购物车中商品件数
- 按照上文的规则 `3`，添加商品的过程有一点点类似于最小堆的sift down过程。

- 从根节点开始，我们选择当前节点的子节点中具有和新增商品相同的最高优先级的节点访问。不幸的是，这不能只通过中间节点的优先级值来判断，因为新增的商品可能和某个节点下的叶子形成新的子树，而这个子树可能具有原来节点更高的优先级。
  - 以*前一个section中的*那张图为例，如果`N1->data == 1`，`N4->data == 2`，`N5->data == 3`，而`L2`和新增节点有相同的最高优先级的跨店折扣(优先级为`0`)，那么在决定访问`N4`还是`L5`时，我们必须访问所有叶子才能知道`L2`可以和新增节点形成新的子树. 这就是假设打折数量较少时O(N)的时间复杂度的来源。如果购物车内商品的折扣数量和购物车内商品可以相比，简单起见假设每个节点下都有m个子节点，那么每访问下一层节点，需要访问的叶子数就减少为上一层的$\frac{1}{m}$, $\sum_{i=0}^{\log_m N} \frac{N}{m^i} = \frac{m}{m-1}(N-1) \approx N$, 时间复杂度还是$O(N)$.
  - 因此，节点优先级分为两种情况：
    1. 若节点的*优先级信息*和新商品相同，则返回所有叶子中的最高优先级
    2. 若节点的*优先级信息*和新商品不同，则返回所有叶子中比当前节点更高的优先级，否则若所有可能形成的新子树的优先级都比当前节点优先级低，就返回一个足够低的优先级(不访问这个节点)
- 添加了节点或者形成新的子树后，子树可能需要往上调整位置，如*下图*第一、二个箭头所示. 这个处理包含在`CartTree::__AddMidNode`函数中。这一步的时间复杂度是$O(1)$。
- 形成新子树且子树向上调整后，若原来的叶子只有1个sibling，如下图第三个树，`L3`没有sibling而且不是root的孩子，因此需要删掉`N5`。这样一来，`L3`甚至可能进入比以前低优先级的另一个子树甚至形成新子树。所以我们将`L3`重新添加回这个树(但是不更新)来保证正确性。之后因为添加时间可能有变化，需要从下到上对每个节点排序。函数`CartTree::__DelBadMidNodes`完成此过程。这一步的时间复杂度是小于$O(N)$的。



- 此外，在每次访问节点时都应该把节点调节到sibling中的最左边，这样从左至右访问叶子恰为满足优先级条件下的按照时间顺序排列. 在函数`CartTree::__AddItem`中实现了这个功能。

删除商品

- 平均时间复杂度: $O(N)$
- 平均空间复杂度: $O(1)$
- 先按照`shop_id`和`item_id`遍历整个树，找到欲删除的商品，这一步的时间复杂度是$O(N)$。如果商品多于1件，只需减少件数即可。否则：
- 若商品有多于1个siblings或者唯一的一个sibling是中间节点，那么树的结构将不需要调整，只要从剩余节点向上更新各个节点的顺序就行。这一步的时间复杂度小于$O(N)$。

    - 这是由于如果删去了在它的sibling中最新的叶子，它的直接parent的添加时间可能会变早，所以要从下到上更新各个节点的`add_time`并用类似冒泡排序的方法向右移动到正确位置。
- 若删掉的商品只有1个sibling且是leaf，那么要处理这个剩余的sibling，和添加商品的步骤类似，只是多判断一步--不仅祖先节点的添加时间可能变得更新，而且还可能变得更旧，因为删掉的那个节点可能正好贡献了最新的添加时间。这一步的时间复杂度小于$O(N)$。

展示购物车

- 递归从左至右遍历叶子节点并调用该`item`的`show`函数即可。
- 额外功能：展示购物车总金额，在遍历到叶子时返回金额，递归调用就容易求得购物车内商品的总金额。

测试和结果

题目给出的测试用例

(从左向右)

1 添加商品

```
Showing Shopping cart:

Total price: 0.00

Adding item: id 1 ...
Add item success

Showing Shopping cart:

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 99.00

Adding item: id 2 ...
Add item success

Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00

Adding item: id 3 ...
Add item success

Showing Shopping cart:

-- Item name: c
   Item id: 3
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 99.00 * 1
-- Item name: b
   Item id: 2
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 286.00

Adding item: id 4 ...
Add item success

Showing Shopping cart:

-- Item name: d
   Item id: 4
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 4
   price: 77.00 * 1

-- Item name: b
   Item id: 2
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

-- Item name: c
   Item id: 3
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 99.00 * 1

Total price: 363.00
```

2 删除商品

```
Showing Shopping cart:

Total price: 0.00

Adding item: id 1 ...
Add item success

Showing Shopping cart:

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 99.00

Adding item: id 2 ...
Add item success

Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00

Adding item: id 3 ...
Add item success

Showing Shopping cart:

-- Item name: c
   Item id: 3
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 99.00 * 1

 -- Item name: a
    Item id: 1
    shop name: 商店Alpha
    Sorry, no discount applicable for this item.
       - 1 pcs in your shopping cart
       -    add time: 1
    price: 99.00 * 1

 -- Item name: b
    Item id: 2
    shop name: 商店Beta
    Sorry, no discount applicable for this item.
       - 1 pcs in your shopping cart
       -    add time: 2
    price: 88.00 * 1

 Total price: 286.00

 deleting item: id 3
 item 3 in shop 0 deleted,
 Showing Shopping cart:

 -- Item name: b
    Item id: 2
    shop name: 商店Beta
    Sorry, no discount applicable for this item.
       - 1 pcs in your shopping cart
       -    add time: 2
    price: 88.00 * 1

 -- Item name: a
    Item id: 1
    shop name: 商店Alpha
    Sorry, no discount applicable for this item.
       - 1 pcs in your shopping cart
       -    add time: 1
    price: 99.00 * 1

 Total price: 187.00
```

3 跨店打折

```
Showing Shopping cart:

Total price: 0.00

Adding item: id 1 ...
Add item success

Showing Shopping cart:

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 折扣1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 99.00

Adding item: id 2 ...
Add item success

Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 折扣1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00

Adding item: id 3 ...
Add item success

Showing Shopping cart:

-- Item name: c
   Item id: 3
   shop name: 商店Gamma
   Discount info: 折扣1-a
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 77.00 * 1


-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 折扣1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

Total price: 264.00

deleting item: id 3
item 3 in shop 2 deleted,
Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 折扣1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00
```

4 跨店满减

```
Showing Shopping cart:

Total price: 0.00

Adding item: id 1 ...
Add item success

Showing Shopping cart:

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 99.00

Adding item: id 2 ...
Add item success

Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00

Adding item: id 3 ...
Add item success

Showing Shopping cart:

-- Item name: c
   Item id: 3
   shop name: 商店Gamma
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 77.00 * 1


-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

Total price: 264.00

deleting item: id 3
item 3 in shop 2 deleted,
Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00
```

5 优先级

```
Showing Shopping cart:

Total price: 0.00

Adding item: id 1 ...
Add item success

Showing Shopping cart:

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 99.00

Adding item: id 2 ...
Add item success

Showing Shopping cart:

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

Total price: 187.00

Adding item: id 4 ...
Add item success

Showing Shopping cart:

-- Item name: d
   Item id: 4
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 66.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

Total price: 253.00

Adding item: id 3 ...
Add item success

Showing Shopping cart:

-- Item name: c
   Item id: 3
   shop name: 商店Gamma
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 4
   price: 77.00 * 1

-- Item name: d
   Item id: 4
   shop name: 商店Alpha
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 3
   price: 66.00 * 1

-- Item name: a
   Item id: 1
   shop name: 商店Alpha
   Discount info: 满减1-a
      - 1 pcs in your shopping cart
      -    add time: 1
   price: 99.00 * 1

-- Item name: b
   Item id: 2
   shop name: 商店Beta
   Sorry, no discount applicable for this item.
      - 1 pcs in your shopping cart
      -    add time: 2
   price: 88.00 * 1

Total price: 330.00
```

额外的测试用例

下面为了便于观察是否正确，打印树结构而不是打印购物车。虽然顺序一样但是这样能展示所有和排序有关的信息。

- 叶子节点中 add time后就是这个物品的最新添加时间，item number是这个物品在购物车内有几件， item id 是商品编号，shop id 是商品所属的店铺编号。
- 节点中的"跨店折扣m-n"意思是第m优先的折扣/满减中第n个。
- 可以观察add time验证是否满足：按照同属一个优先级的最新节点排序

1 多个不同优先级折扣

图中可以看出`跨店折扣2-a`的优先级比同商店更大，但是比`跨店折扣1-a`和`跨店折扣1-b`更小。

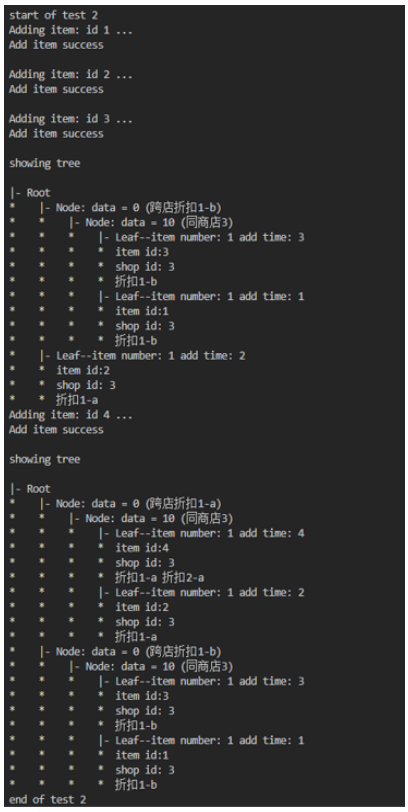
```
Adding item: id 1 ...
Add item success

Adding item: id 2 ...
Add item success

Adding item: id 3 ...
Add item success

Adding item: id 4 ...
Add item success

Adding item: id 5 ...
Add item success

Adding item: id 6 ...
Add item success

Adding item: id 7 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 7 ...
Add item success

Adding item: id 7 ...
Add item success

Adding item: id 7 ...
Add item success

Adding item: id 8 ...
Add item success

Adding item: id 9 ...
Add item success

Adding item: id 10 ...
Add item success

Adding item: id 13 ...
Add item success

Adding item: id 4 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 11 ...
Add item success

Adding item: id 14 ...
Add item success

showing tree
```

```
|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Node: data = 10 (同商店1)
*    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    |- Leaf--item number: 1 add time: 20
*    *    *    *    *  item id:14
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-b 满减1-a
*    *    *    *    |- Leaf--item number: 3 add time: 18
*    *    *    *    *  item id:12
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-b 满减1-a
*    *    *    *    |- Leaf--item number: 1 add time: 15
*    *    *    *    *  item id:10
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-b 满减1-a
*    *    *    |- Node: data = 1 (跨店折扣2-a)
*    *    *    *    |- Node: data = 10 (同商店1)
*    *    *    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    *    *    |- Leaf--item number: 1 add time: 19
*    *    *    *    *    *    *  item id:11
*    *    *    *    *    *    *  shop id: 1
*    *    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
*    *    *    *    *    *    |- Leaf--item number: 1 add time: 16
*    *    *    *    *    *    *  item id:13
*    *    *    *    *    *    *  shop id: 1
*    *    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
*    *    *    |- Leaf--item number: 1 add time: 5
*    *    *    *  item id:5
*    *    *    *  shop id: 5
*    *    *    *  折扣1-b
*    *    *    |- Node: data = 10 (同商店2)
*    *    *    *    |- Leaf--item number: 1 add time: 4
*    *    *    *    *  item id:4
*    *    *    *    *  shop id: 2
*    *    *    *    *  折扣1-b
*    *    *    *    |- Leaf--item number: 1 add time: 3
*    *    *    *    *  item id:3
*    *    *    *    *  shop id: 2
*    *    *    *    *  折扣1-b 满减1-a
*    *    |- Node: data = 0 (跨店折扣1-a)
*    *    *    |- Leaf--item number: 1 add time: 17
*    *    *    *  item id:4
*    *    *    *  shop id: 3
*    *    *    *  折扣1-a 折扣2-a
*    *    *    |- Node: data = 10 (同商店2)
*    *    *    *    |- Node: data = 11 (跨店满减1-b)
*    *    *    *    *    |- Leaf--item number: 1 add time: 14
*    *    *    *    *    *  item id:9
*    *    *    *    *    *  shop id: 2
*    *    *    *    *    *  折扣1-a 满减1-b
*    *    *    *    *    |- Leaf--item number: 4 add time: 12
*    *    *    *    *    *  item id:7
*    *    *    *    *    *  shop id: 2
*    *    *    *    *    *  折扣1-a 满减1-b
*    *    *    *    |- Leaf--item number: 1 add time: 2
*    *    *    *    *  item id:2
*    *    *    *    *  shop id: 2
*    *    *    *    *  折扣1-a
*    *    *    |- Node: data = 10 (同商店1)
*    *    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    *    |- Leaf--item number: 1 add time: 13
*    *    *    *    *    *  item id:8

*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-a 满减1-a
*    *    *    *    |- Leaf--item number: 1 add time: 1
*    *    *    *    *  item id:1
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-a 满减1-a
*    *    |- Leaf--item number: 1 add time: 6
*    *    *  item id:6
*    *    *  shop id: 1
*    *    *  折扣1-c
end of test 1
```

2 形成新子树(1)

从较小优先级的节点里夺走叶子，形成新的子树

```
/* 加几个discount2 1-2形成一个中间节点(各个discount1
1-1不同)，然后再加一个
 * discount1 1-1 和其中一个一样的，形成新的中间节点
 *
 *          root_                        root_
 *           |                           /     \
 *          mid_2  _           ->      mid_2      mid_3  _
 *         /   \     \                   |  \        |      \
 *      leaf_1 leaf_2 leaf_3       leaf_1 leaf_3   leaf_2  new_leaf
 *
 * */
```



```
start of test 2
Adding item: id 1 ...
Add item success

Adding item: id 2 ...
Add item success

Adding item: id 3 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Node: data = 10 (同商店3)
*    *    *    |- Leaf--item number: 1 add time: 3
*    *    *    *  item id:3
*    *    *    *  shop id: 3
*    *    *    *  折扣1-b
*    *    *    |- Leaf--item number: 1 add time: 1
*    *    *    *  item id:1
*    *    *    *  shop id: 3
*    *    *    *  折扣1-b
*    |- Leaf--item number: 1 add time: 2
*    *  item id:2
*    *  shop id: 3
*    *  折扣1-a
Adding item: id 4 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-a)
*    *    |- Node: data = 10 (同商店3)
*    *    *    |- Leaf--item number: 1 add time: 4
*    *    *    *  item id:4
*    *    *    *  shop id: 3
*    *    *    *  折扣1-a 折扣2-a
*    *    *    |- Leaf--item number: 1 add time: 2
*    *    *    *  item id:2
*    *    *    *  shop id: 3
*    *    *    *  折扣1-a
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Node: data = 10 (同商店3)
*    *    *    |- Leaf--item number: 1 add time: 3
*    *    *    *  item id:3
*    *    *    *  shop id: 3
*    *    *    *  折扣1-b
*    *    *    |- Leaf--item number: 1 add time: 1
*    *    *    *  item id:1
*    *    *    *  shop id: 3
*    *    *    *  折扣1-b
end of test 2
```

3 形成新子树(2)

形成的新子树可能有不止一个节点的深度，除了新形成的节点外，原有的节点也要添加到正确的位置。

```
start of test 3
Adding item: id 13 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 12 ...
Add item success

Adding item: id 11 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Node: data = 1 (跨店折扣2-a)
*    *    *    |- Node: data = 10 (同商店1)
*    *    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    *    |- Leaf--item number: 1 add time: 5
*    *    *    *    *    *  item id:11
*    *    *    *    *    *  shop id: 1
*    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
*    *    *    *    *    |- Leaf--item number: 1 add time: 1
*    *    *    *    *    *  item id:13
*    *    *    *    *    *  shop id: 1
*    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
*    *    |- Leaf--item number: 1 add time: 4
*    *    *  item id:12
*    *    *  shop id: 1
*    *    *  折扣1-b 满减1-a
Adding item: id 14 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Node: data = 10 (同商店1)
*    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    |- Leaf--item number: 1 add time: 6
*    *    *    *    *  item id:14
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-b 满减1-a
*    *    *    *    |- Leaf--item number: 1 add time: 4
*    *    *    *    *  item id:12
*    *    *    *    *  shop id: 1
*    *    *    *    *  折扣1-b 满减1-a
*    *    |- Node: data = 1 (跨店折扣2-a)
*    *    *    |- Node: data = 10 (同商店1)
*    *    *    *    |- Node: data = 11 (跨店满减1-a)
*    *    *    *    *    |- Leaf--item number: 1 add time: 5
*    *    *    *    *    *  item id:11
*    *    *    *    *    *  shop id: 1
*    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
*    *    *    *    *    |- Leaf--item number: 1 add time: 1
*    *    *    *    *    *  item id:13
*    *    *    *    *    *  shop id: 1
*    *    *    *    *    *  折扣1-b 折扣2-a 满减1-a
end of test 3
```

4 形成新子树(3)

形成新子树后，原节点的添加时间可能发生改变，要正确重新排序。可以看到这里item 2 如何被正确地排到item 14 的后面

```
start of test 4
Adding item: id 2 ...
Add item success

Adding item: id 14 ...
Add item success

Adding item: id 1 ...
Add item success

showing tree

|- Root
*    |- Node: data = 10 (同商店3)
*    *    |- Leaf--item number: 1 add time: 3
*    *    *  item id:1
*    *    *  shop id: 3
*    *    *  折扣1-b
*    *    |- Leaf--item number: 1 add time: 1
*    *    *  item id:2
*    *    *  shop id: 3
*    *    *  折扣1-a
*    |- Leaf--item number: 1 add time: 2
*    *  item id:14
*    *  shop id: 14
*    *
Adding item: id 3 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Leaf--item number: 1 add time: 4
*    *    *  item id:3
*    *    *  shop id: 2
*    *    *  折扣1-b 满减1-a
*    *    |- Leaf--item number: 1 add time: 3
*    *    *  item id:1
*    *    *  shop id: 3
*    *    *  折扣1-b
*    |- Leaf--item number: 1 add time: 2
*    *  item id:14
*    *  shop id: 14
*    *
*    |- Leaf--item number: 1 add time: 1
*    *  item id:2
*    *  shop id: 3
*    *  折扣1-a
end of test 4
```

5 形成新子树(4)

形成新子树剩下了一个节点时，要正确地去除剩余节点上方的没有sibling的节点。如 `规则4` 所述。

可以注意到item 1 形成新子树后，item 2 正确地去除了上方的节点，成为 root的孩子。

```
start of test 5
Adding item: id 1 ...
Add item success

Adding item: id 2 ...
Add item success

Adding item: id 15 ...
Add item success

showing tree

|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:15
*    *  shop id: 15
*    *
*    |- Node: data = 10 (同商店3)
*    *    |- Leaf--item number: 1 add time: 2
*    *    *  item id:2
*    *    *  shop id: 3
*    *    *  折扣1-a
*    *    |- Leaf--item number: 1 add time: 1
*    *    *  item id:1
*    *    *  shop id: 3
*    *    *  折扣1-b
Adding item: id 3 ...
Add item success

showing tree

|- Root
*    |- Node: data = 0 (跨店折扣1-b)
*    *    |- Leaf--item number: 1 add time: 4
*    *    *  item id:3
*    *    *  shop id: 2
*    *    *  折扣1-b 满减1-a
*    *    |- Leaf--item number: 1 add time: 1
*    *    *  item id:1
*    *    *  shop id: 3
*    *    *  折扣1-b
*    |- Leaf--item number: 1 add time: 3
*    *  item id:15
*    *  shop id: 15
*    *
*    |- Leaf--item number: 1 add time: 2
*    *  item id:2
*    *  shop id: 3
*    *  折扣1-a
end of test 5
```

6 删除只有1个sibling的叶子节点 (1)

可能由于被删节点贡献了上方某些节点的最新添加时间，删除之后的这些节点需要重新排序，如下图

```
/**
 *                root                                root
 *                  |                                   |
 *             mid_node1  _              (del leaf1)   mid_node1
 *           /      |         \             ->           |     \
 *      mid_node2  leaf3  mid_node3 (shop2)              mid_node3   \
 *    /(discount1)\         /       \                    /    |  \     leaf3
 *  leaf1      leaf2     leaf4    leaf5             leaf2  leaf4  leaf5
 *  (shop1)   (shop2)                        add time:  2      4    5     3
 */
```

可以看到item 3被排到正确的位置

```
start of test 6
Adding item: id 2 ...
Add item success

Adding item: id 1 ...
Add item success

Adding item: id 25 ...
Add item success

Adding item: id 6 ...
Add item success

Adding item: id 23 ...
Add item success

showing tree

|- Root
*    |- Node: data = 10 (同商店1)
*    *    |- Leaf--item number: 1 add time: 5
*    *    *  item id:23
*    *    *  shop id: 1
*    *    *
*    *    |- Leaf--item number: 1 add time: 4
*    *    *  item id:6
*    *    *  shop id: 1
*    *    *  折扣1-c
*    |- Leaf--item number: 1 add time: 3
*    *  item id:25
*    *  shop id: 100
*    *
*    |- Node: data = 0 (跨店折扣1-a)
*    *    |- Leaf--item number: 1 add time: 2
*    *    *  item id:1
*    *    *  shop id: 1
*    *    *  折扣1-a 满减1-a
*    *    |- Leaf--item number: 1 add time: 1
*    *    *  item id:2
*    *    *  shop id: 2
*    *    *  折扣1-a

deleting item: id 2
item 2 in shop 2 deleted,
after del
|- Root
*    |- Node: data = 10 (同商店1)
*    *    |- Leaf--item number: 1 add time: 5
*    *    *  item id:23
*    *    *  shop id: 1
*    *    *
*    *    |- Leaf--item number: 1 add time: 4
*    *    *  item id:6
*    *    *  shop id: 1
*    *    *  折扣1-c
*    *    |- Leaf--item number: 1 add time: 2
*    *    *  item id:1
*    *    *  shop id: 1
*    *    *  折扣1-a 满减1-a
*    |- Leaf--item number: 1 add time: 3
*    *  item id:25
*    *  shop id: 100
*    *
end of test 6
```

7 删除到空树

```
start of test 7
Adding item: id 17 ...
Add item success

Adding item: id 17 ...
Add item success

Adding item: id 14 ...
Add item success

before del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:14
*    *  shop id: 14
*    *
*    |- Leaf--item number: 2 add time: 2
*    *  item id:17
*    *  shop id: 111
*    *  满减1-a
deleting item: id 17
item 17 in shop 111 deleted, 1pcs of the same item remaining
after del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:14
*    *  shop id: 14
*    *
*    |- Leaf--item number: 1 add time: 2
*    *  item id:17
*    *  shop id: 111
*    *  满减1-a
before del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:14
*    *  shop id: 14
*    *
*    |- Leaf--item number: 1 add time: 2
*    *  item id:17
*    *  shop id: 111
*    *  满减1-a
deleting item: id 17
item 17 in shop 111 deleted,
after del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:14
*    *  shop id: 14
*    *
before del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:14
*    *  shop id: 14
*    *
deleting item: id 14
item 14 in shop 14 deleted (last item in cart)
after del
|- Root
end of test 7
```

8 删除根节点下的商品

```
start of test 8
Adding item: id 14 ...
Add item success

Adding item: id 15 ...
Add item success

Adding item: id 16 ...
Add item success

before del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:16
*    *  shop id: 16
*    *
*    |- Leaf--item number: 1 add time: 2
*    *  item id:15
*    *  shop id: 15
*    *
*    |- Leaf--item number: 1 add time: 1
*    *  item id:14
*    *  shop id: 14
*    *
deleting item: id 15
item 15 in shop 15 deleted,
after del
|- Root
*    |- Leaf--item number: 1 add time: 3
*    *  item id:16
*    *  shop id: 16
*    *
*    |- Leaf--item number: 1 add time: 1
*    *  item id:14
*    *  shop id: 14
*    *
end of test 8
```

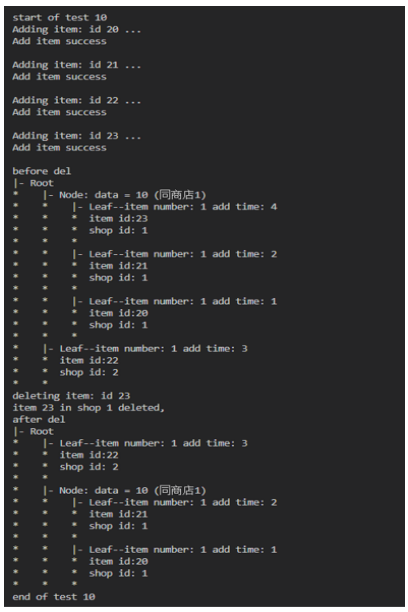
9 删除有多件相同商品中的一件

这不会改变展示顺序

```
start of test 9
Adding item: id 14 ...
Add item success

Adding item: id 15 ...
Add item success

Adding item: id 16 ...
Add item success

Adding item: id 15 ...
Add item success

Adding item: id 16 ...
Add item success

Adding item: id 15 ...
Add item success

Adding item: id 16 ...
Add item success

before del
|- Root
*    |- Leaf--item number: 3 add time: 7
*    *  item id:16
*    *  shop id: 16
*    *
*    |- Leaf--item number: 3 add time: 6
*    *  item id:15
*    *  shop id: 15
*    *
*    |- Leaf--item number: 1 add time: 1
*    *  item id:14
*    *  shop id: 14
*    *
deleting item: id 15
item 15 in shop 15 deleted, 2pcs of the same item remaining
after del
|- Root
*    |- Leaf--item number: 3 add time: 7
*    *  item id:16
*    *  shop id: 16
*    *
*    |- Leaf--item number: 2 add time: 6
*    *  item id:15
*    *  shop id: 15
*    *
*    |- Leaf--item number: 1 add time: 1
*    *  item id:14
*    *  shop id: 14
*    *
end of test 9
```

10 删除某个大于2个sibling的叶子节点

虽然无需删除剩余节点上方只有1个孩子的节点，但是仍然要对剩余节点进行重新排序，因为可能删除了的节点贡献了最新的添加时间。可以看到由于item 23 被删除，`同商店1` node的时间从早于，item 22变为晚于item 22

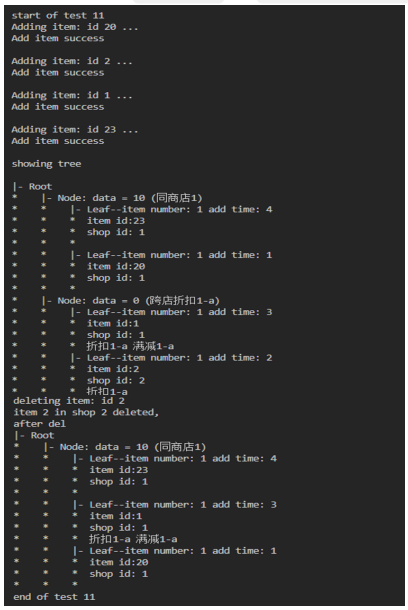


11 删除只有1个sibling的叶子节点 (2)

剩余的叶子在删除上方独支之后可能进入旁边已有子树。

```
    /**
     *                root                              root
     *                  |                                 |
     *              mid_node1            (del leaf1)   mid_node1
     *           /           \              ->                \
     *      mid_node2    mid_node3 (shop2)                    mid_node3
     *    /(discount1)\     /       \                         /    |   \
     *  leaf1      leaf2  leaf3    leaf4                   leaf3  leaf2  leaf4
     *  (shop1)   (shop2)                          add time:  1      3       4
     */
```

可以看到`leaf2`从`mid_node2`进入`mid_node3`



1. *优先级信息*和*优先级*是不同的。比如，某个中间节点的优先级对应着跨店折扣中第二优先的那些折扣中的第三个，那么就要求它的所有后代叶子都有第二优先的那些折扣中的第三个。上述中间节点的优先级是全局第二高的，但它和不同*优先级信息* (折扣/满减/跨店) 的节点是不同的，即使有相同优先级。因为题目要求商城中有多个满减/折扣，这是自然的。 ↩
2. 这对应着购物车排序中以"某个折扣或满减中最新的商品"的添加时间进行排序的要求 ↩
3. 事实上，这相当于从上到下逐个访问它的祖先节点--正如规则`1`要求的，"辈分最高"的祖先节点一定有最高的优先级 ↩
4. 我们定义优先级越大，x_i越小，所以$\{x_i\}$是降序。 ↩
5. 这对应着先满足高优先级的商品相邻展示，后满足低优先级的商品相邻展示的要求。一种冗长的描述是：某个叶子的正确位置的祖先最高优先级应该不低于其他位置的祖先的最高优先级，如果这里是等号，就要接下来比较第二高优先级，循环上述操作；如果不是等号，那就说明正确位置一定比这个位置好。此外如果在一个位置的sibling中有这样一个节点，它的优先级 ↩
6. 我们要求一个商品不能有相同优先级的不同折扣，否则排序将无法定义 ↩
7. 事实上不用线性表也行，因为折扣信息之间没关系，只要能通过地址索引到就可以 ↩