

练习题

- 部署一个可升级的 NFT 市场合约
 - 第一版本普通合约
 - 第二版本，加入离线签名上架 NFT 功能方法（签名内容：tokenId， 价格），实现用户一次性 setApproveAll 给 NFT 市场合约，每次上架时使用签名上架。
- 部署到测试网，并开源到区块链浏览器（在 Readme.md 中备注代理合约及两个实现的合约地址）

升级的思考

- 优点：

- 缺点：

Gas 技巧

- 区分 交易 **Gas** 和 部署 **Gas**
- 修改变量顺序 -> 合并槽（但尽量使用 uint256 的变量）
- 常量或 immutable 代替变量
- 精确声明Solidity合约函数的可见性
- 避免无限制的循环
- 合约中没有引用的变量，用事件
- 减少链上数据：IPFS 链下存储
- 使用代理进行大规模部署（复用实现合约）
- 链下计算、链上验证（如：数组用链表实现、可迭代的链表）
- 在合约验证数据，而不是存储数据（如使用 **Merkle** 树）

考虑需求

创建一个“学校”智能合约来收集学生地址。合约必须具有3个主要功能：

1. 在合约中添加或删除学生。
2. 询问给定的学生地址是否属于学校。
3. **获取所有学生的名单。**

Demo

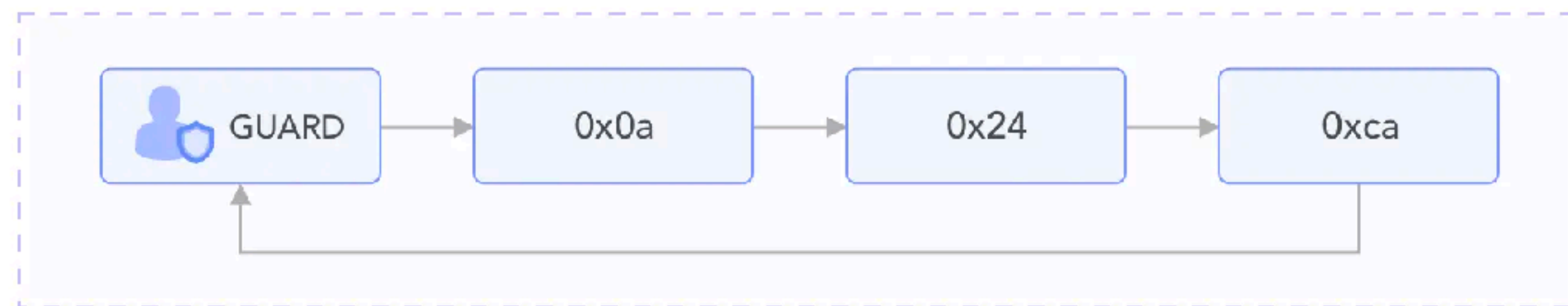
Mapping实现: SchoolMapping.sol

数组实现: SchoolBaseArray.sol

Gas 对比

Total number student in list	10 students		100 students	
	mapping	array	mapping	array
add	43,704	57,064	43,704	138,364
remove	13,721	59,612	13,721	416,662

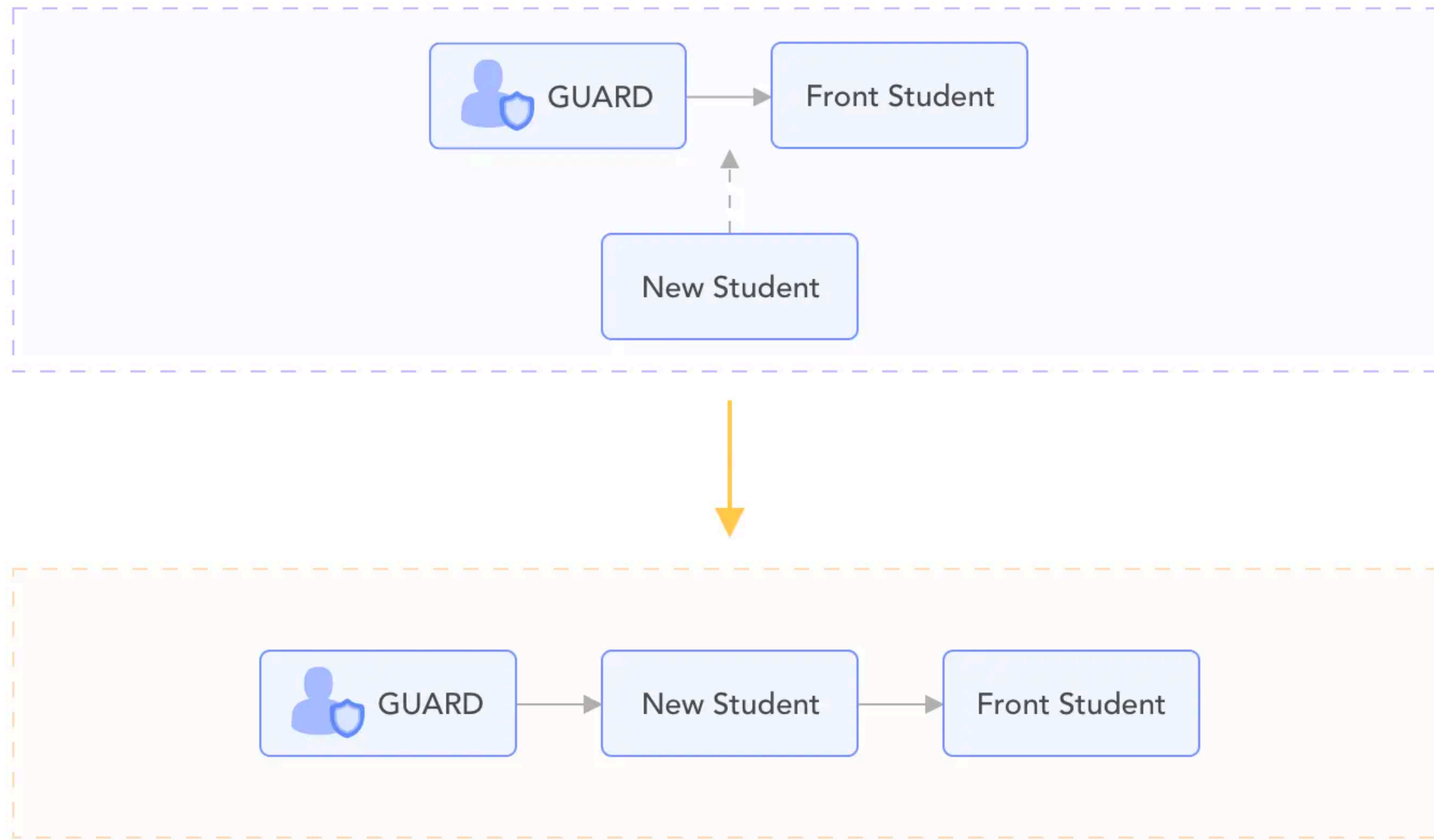
可迭代链表



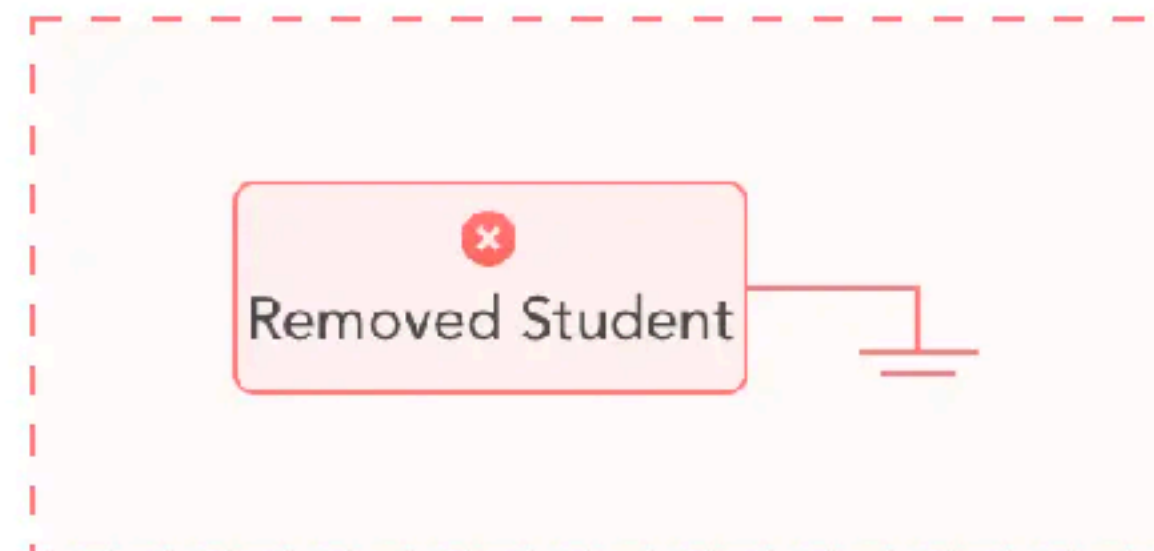
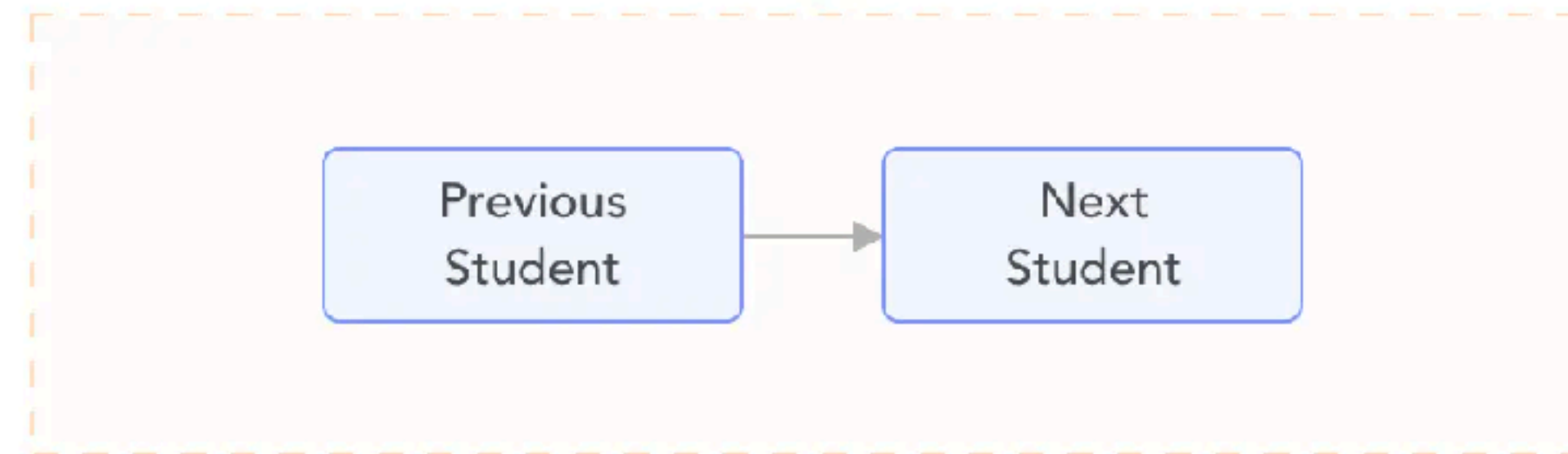
Key	Value
GUARD	0x0a
0x0a	0x24
0x24	0xca
0xca	GUARD

Mapping

可迭代链表 – 添加



可迭代链表 – 删除



可迭代链表 – 性能

Total number student in list	10 students	100 students
add	54,667	54,667
remove (unoptimized)	31,680	83,040
remove (optimized)	23,974	23,974

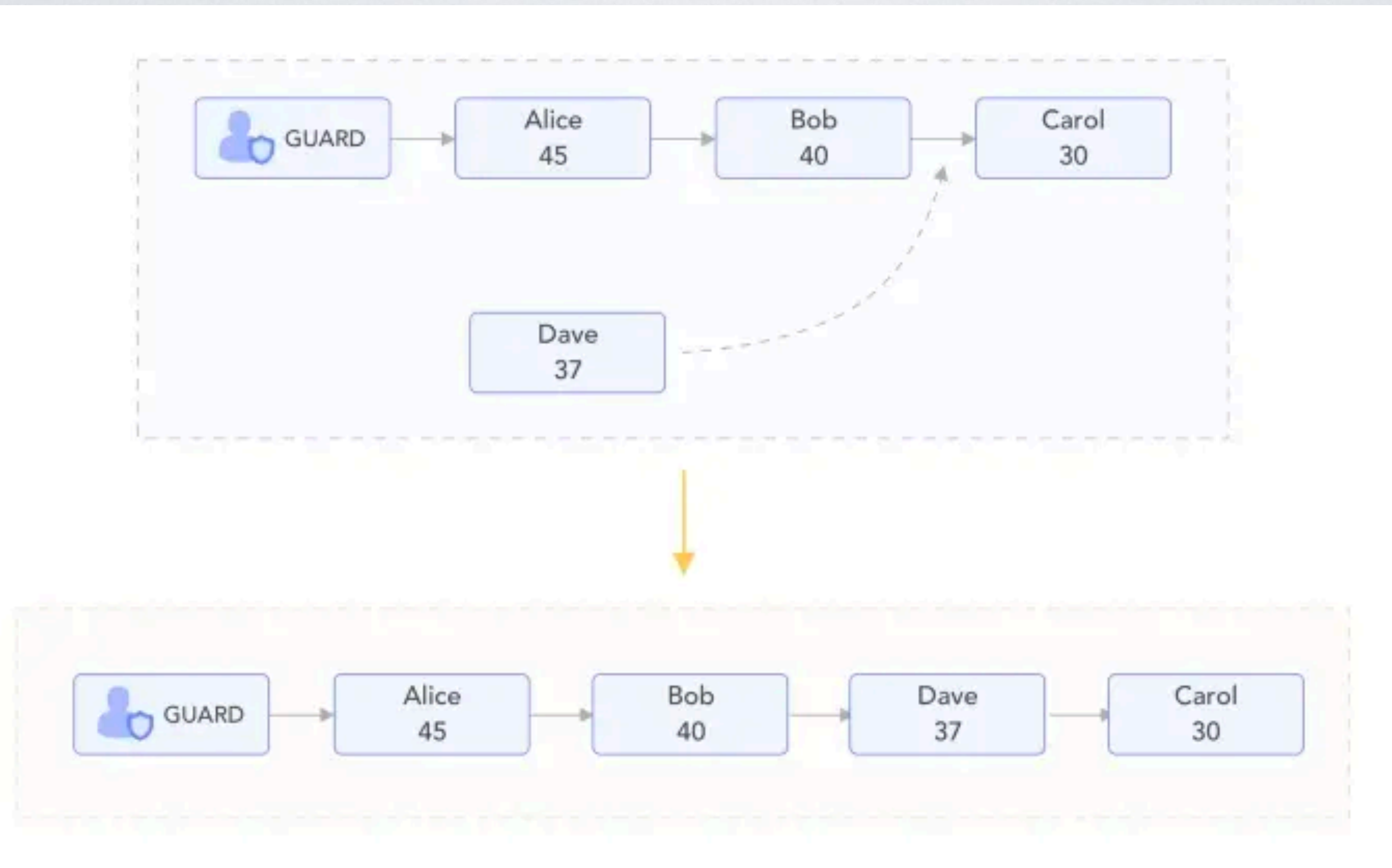
SchoolMappingList.sol

考虑新需求

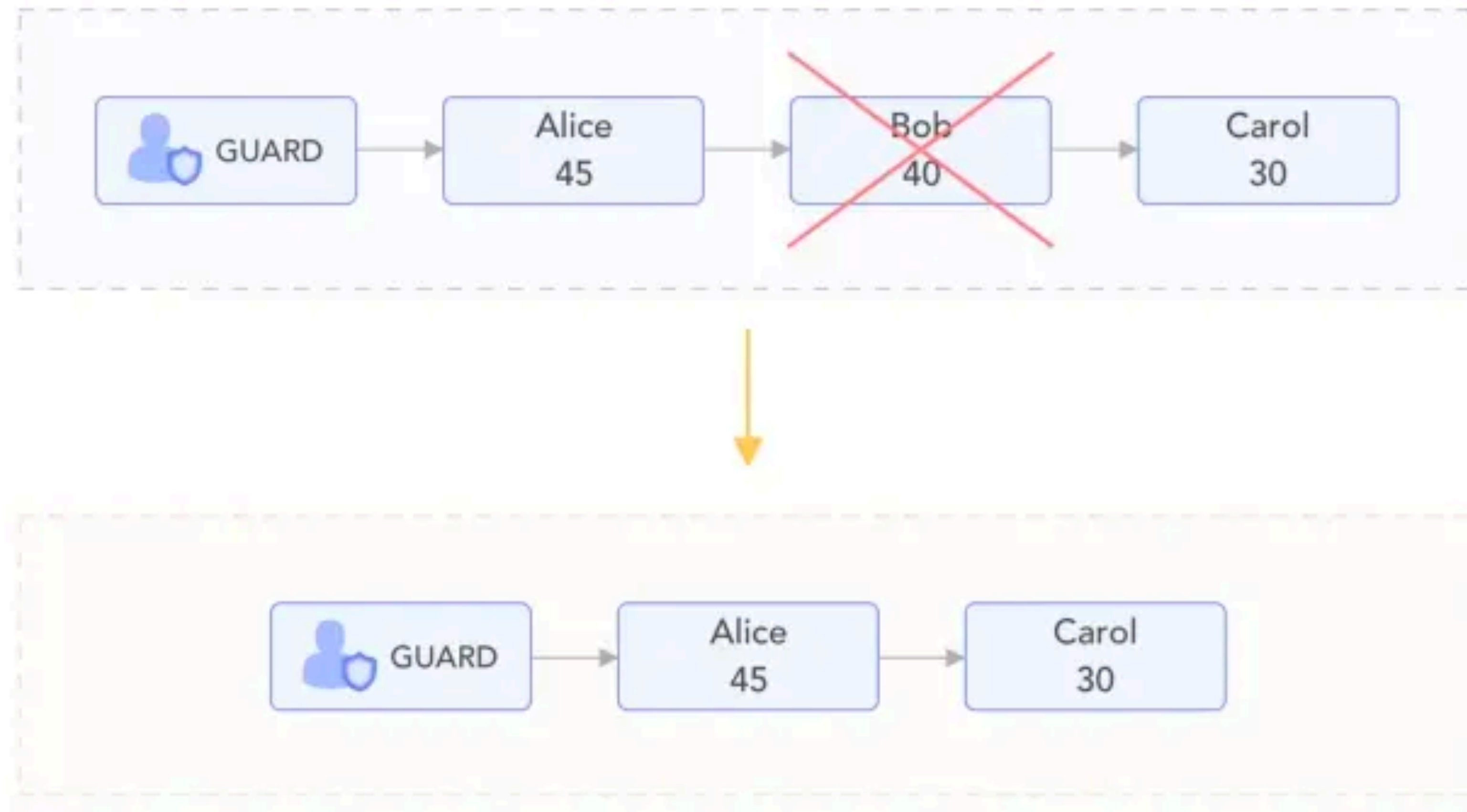
需要根据分数来维持学生的**排序**，功能需求如下：

1. 将新学生添加到根据分数排序的列表中
2. 提高学生分数（保持列表排序）
3. 降低学生分数（保持列表排序）
4. 从名单中删除学生
5. **获取前K名学生名单**

可迭代链表 – 有序插入



可迭代链表 – 列表删除



可迭代链表 – 更新分数



优化的可迭代链表

[https://github.com/OpenSpace100/blockchain-tasks/blob/main/solidity_sample_code/
SchoolOptimized.sol](https://github.com/OpenSpace100/blockchain-tasks/blob/main/solidity_sample_code/SchoolOptimized.sol)

练习题

- 之前的 TokenBank 合约的前 3 名，修改为前 10 名，并用可选代的链表保存。

<https://decert.me/challenge/753d5050-e5e4-4a0d-8ad9-9ecd7e0e0788>

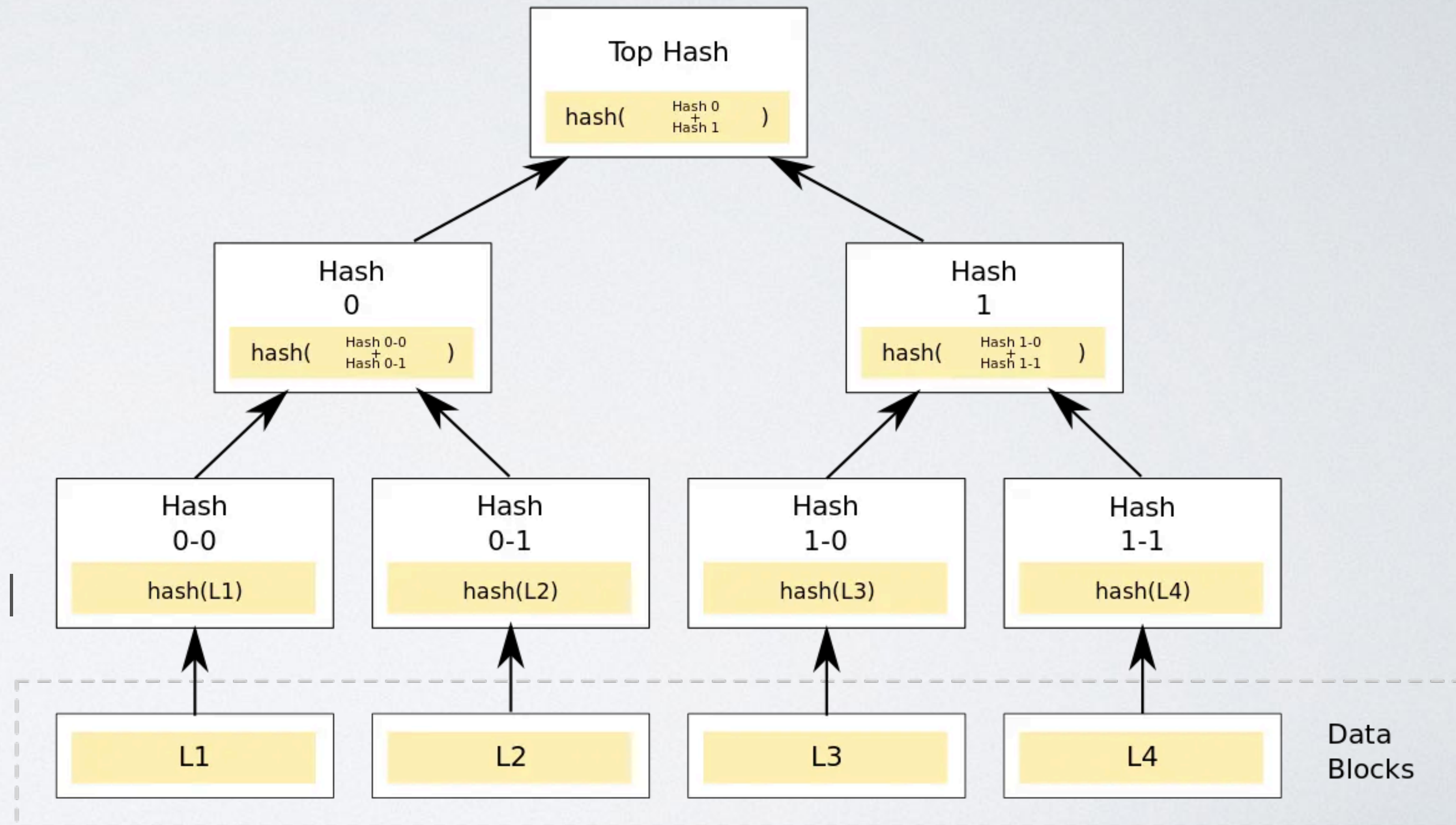
白名单实现

- 映射: `mapping(address => bool) public whitelist;`
- 后台离线签名, 省 gas , 比较灵活 (需要从后端获取)
- 默克尔树 (Merkle Tree)

默克尔树

构建好树之后，
验证证明只需要对数级的时间！

验证 L1 只需要 Hash0-1 和 Hash 1



默克尔树

- 构建白名单默克尔树 (Merkle Tree)
 - https://github.com/OpenSpace100/blockchain-tasks/tree/main/merkle_distributor
- 在合约里验证是在默克尔树中
 - https://github.com/OpenSpace100/blockchain-tasks/blob/main/solidity_sample_code/MerkleAirdrop.sol

MultiCall

- 每次交易至少需要21000 gas，可否多个交易封在一个交易里
- 如何在同一个交易里同时调用合约里的多个（次）函数？

MultiCall 写

```
function multicall(bytes[] calldata data) external virtual returns (bytes[]  
memory results) {  
    results = new bytes[](data.length);  
    for (uint256 i = 0; i < data.length; i++) {  
        results[i] = this.delegateCall(data[i]);  
    }  
    return results;  
}
```


MULTICALL – 读

- 打包读取和打包写入， 在一次 RPC 请求封装多个请求
 - 降低网络请求
 - 保证数据来自一个区块
 - Viem 有原生集成： <https://viem.sh/docs/contract/multicall.html>
- 打包写入， 仅使用于不关注 msg.sender 情况。

<https://github.com/mdsl/multicall/blob/main/src/Multicall3.sol>

MultiCall – 读

```
function aggregate3(Call3[] calldata calls) public payable returns (Result[] memory returnData) {
    uint256 length = calls.length;
    returnData = new Result[](length);
    Call3 calldata calli;
    for (uint256 i = 0; i < length;) {
        Result memory result = returnData[i];
        calli = calls[i];
        (result.success, result.returnData) = calli.target.call(calli.callData);
        assembly {
            // ...
        }
        unchecked { ++i; }
    }
}
```


练习题

- 组合使用 Multicall (delegateCall) 、默克尔树、erc20 permit 授权，实现一个默克尔树优惠价格（100Token）购买名单，使用 Multicall. 调用 2 个方法：
 - 合约：AirdopMerkleNFTMarket:
 - permitPrePay：完成预授权
 - claimNFT()：验证白名单，如果通过，直接使用 permitPrePay 的授权，转入 100 token，并转出 NFT

<https://decert.me/quests/faa435a5-f462-4f92-a209-3a7e8fdc4d81>