1. 管理员调用函数发送代币

在这种场景下，所有者必须支付gas费才能调用该函数，如果地址列表很大，尤其是在ETH上，这将是不可持续的。

2. 在合约上存储白名单地址列表

您可能会实现一个映射，该映射 mapping(address => some struct)存储所有列入白名单的地址以及该地址是否已认领空投。同样，所有者也必须支付gas费用来存储合约的白名单地址列表。

3.Merkle空投

对于Merkle空投，实现了相同的目标并具有以下好处：

所有者只需支付gas费来创建合约并将 Merkle 根存储在合约上。

列入白名单的地址可以自行调用合约来申领空投——这也开启了在截止日期前申领空投的可能性。

创建Merkle空投的步骤

代码参考可以在 https://github.com/steve-ng/merkle-airdrop 找到——使用了 2 个主要库

前端：<https://github.com/miguelmota/merkletreejs>

Solidity:<https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v4.3/contracts/utils/cryptography/MerkleProof.sol>

图示

描述已自动生成图示

描述已自动生成

先决条件

生成他们有资格获得的白名单和金额列表

根据列表生成Merkle根

<https://github.com/steve-ng/merkle-airdrop/blob/main/test/MerkleDistributor.ts>

智能合约部分

生成的 Merkle根存储在你的智能合约中——可以参考 <https://github.com/steve-ng/merkle-airdrop/blob/main/contracts/MerkleDistributor.sol>

前端部分

存储所有符合空投条件的地址，这样当用户访问你的站点时，他们可以立即查看他们是否符合条件

如果他们符合条件，请使用证明调用智能合约。

参考<https://github.com/steve-ng/merkle-airdrop/blob/main/test/MerkleDistributor.ts#L46>