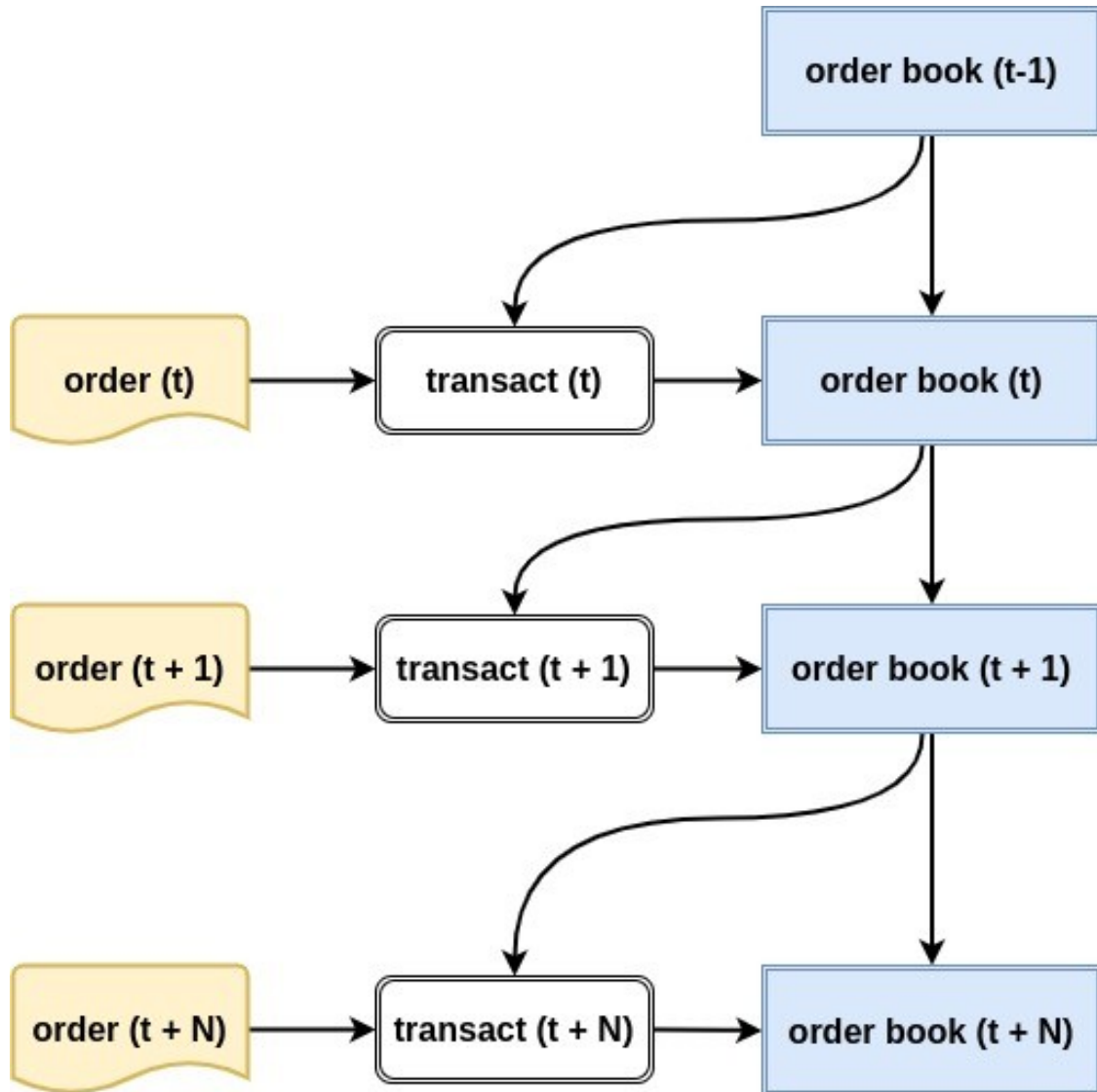


N 档订单簿

在常见的交易软件中，都会实时展示股票的买卖报单盘口行情，通常为 N 个当前最优的待成交“买”/“卖”报价单的委托价格及数量，我们称之为 N 档订单簿（N-Order Book）。N 档订单簿的实时生成方式为：接收最新的报价单，与当前的订单簿按照订单优先级进行撮合成交，并更新撮合后的 N 档订单簿，其流程如下：



订单数据字段说明

- `idx (int)`: 报单编号
- `time (str)`: 报单时间
- `price (float)`: 委托价格

- volume (int): 委托数量
- quote_type (str): 报单类型, BID -> 委托买单; ASK -> 委托卖单
- order_type (str): 订单类型, LIMIT -> 限价单; MARKET -> 市价单

订单撮合成交规则

- 订单簿成交优先级：[价格优先 > 时间优先 > 报单编号优先]
- 委托方式：
 - 限价单：指定可接受的最坏成交价格（若为买单，则 \leq 该价格均可成交；若为卖单，则 \geq 该价格均可成交）
 - 市价单：订单到达交易所时，会以对手价 1 档申报价格（若为买单，则对手价 1 档为当前订单簿卖单中的最低价格；若为卖单，则对手价 1 档为当前订单簿买单中的最高价格）
- 撮合方式：价格成交条件: 若委托买单，则“对手单价格 \leq 委托价格”均可成交；若委托卖单，则“对手单价格 \geq 委托价格”均可成交. 将满足价格成交条件的对手单按照订单簿优先级排序逐一进行委托数量的比对，数量小的订单算作完全成交，移出订单簿；数量大的将登记入订单簿。

题目要求

```
from abc import (
    ABCMeta,
    abstractmethod
)
from typing import Any
```

```
class BaseBroker(object, metaclass=ABCMeta):
```

```
    @abstractmethod
    def transact(self, order: Any, **kwargs):
        """
        Transact: deal with each order sequentially.

        Args:
            order (Any): Required.
        """
        raise NotImplementedError

    @abstractmethod
    def order_book(self, level: int = 5, **kwargs):
        """
        Order book: N level.

        Args:
            level (int): Optional, N-level, default is 5.
```

```
"""  
raise NotImplementedError
```

请实现一个可生成 N 档订单簿的 **Broker** (如上为基类), 并尽可能使得代码的运行耗时最低。