

Liquidity Iteration

When swapping through liquidity via a Swap, Multi-Hop Swap, or a Taker Limit Order we iterate through the availableTickLiquidity to fill the swap order. Liquidity is always iterated through in order of best to worst price. In the case of swapping Token0 (tokenIn) for Token1 (tokenOut) we iterate through tick indexes left to right (eg. -1, 0, 1, 2...) and for Token1 for Token0 we iterate from right to left (eg. 2, 1, 0, -1...) For each swap we completely exhaust the available reserves before moving onto the next tick. ForTickLiquidity instances at the sameTickIndex they are iterated through in a deterministic order as follows:

- PoolReserves
- : In AscendingFee
- order
- LimitOrderTranches
- : In ascendingTrancheKey
- order

When swapping throughPoolReserves the proceeds from the swap are added to the reserves on the reciprocal side of the pool. Ie. The output of TokenA swapped through aPoolReserves at tick 1, fee 1 will moved to aPoolReserves holding TokenB at tick -1 fee 1.

Example liquidity iteration swapping TokenA for TokenB

1. The first availableTickLiquidity
2. holding token TokenB is aPoolReserves
3. at tick 5, fee 1. When this is swapped through the TokenIn is deposited in thePoolReserves
4. at tick 3.
5. The poolPoolReserves
6. at tick 5, fee 2 is swapped through. The TokenIn is deposited in thePoolReserves
7. at tick 2.
8. The limit order at tick 5 is swapped through.
9. ThePoolReserves
10. at tick 6, fee 1 is swapped through. The TokenIn is deposited in thePoolReserves
11. at tick 4. [Previous Tick Liquidity Next Swaps](#)