Non-Performant Algorithm

Quant: Ethan
Desk: Ethan Jerry Cao
Backtester:
Implementer:

Introduction:

Implementation of a very simple trend strategy: if the stock reverses its trend, we will swap our position accordingly. This will by highly non-performant and is written primarily for the sake of constructing a viable structure for receiving market updates & sending buy/sell orders.

Note: please see this implementation in Desk-Alpha/Strats/Template/MultipleStockTemplate.py in the jerrycao-dev fork.

Inputs:

- This algorithm only requires standard ticker: price data.
- The input data structure should be a dictionary of key[ticker] and value[stock data], where each element represent the data on one single stock (given by stocksdata[ticker])
    - Each stock's data is also a dictionary, where the keys are "price" and "trend"
    - "trend" of each stock is initialized to "downwards" so once price goes up for the first time we buy the stock
    - The method "update_all(self, new_stocksdata) automatically updates each stock's "trend" based on its previous and new price. User doesn't need to worry about updating the trend, just pass in the stock prices!

Outputs:

- Upon calling "update_all(self, new_stocksdata)", a dictionary is returned where key is the stock ticker. The value associated with each key (ticker) is a list of orders (list of "BUY" and "SELL")

Backtesting:

Pending, but this algorithm will be terrible.