```
import requests
import signal
from time import sleep


class ApiException(Exception):
   pass

def signal_handler(signum, frame):
    global shutdown
    signal.signal(signal.SIGINT, signal.SIG_DFL)
    shutdown = True


API_KEY = {'X-API-key': 'GSV8A2SJ'}
shutdown = False

#change variables
SPREAD = 0.2
M_SPREAD = 0.05
RISK1 = 0.01
SPEEDBUMP = 0.25
alpha = 0.01
BUMP=0.01
UB_pos = 0
GEM_pos = 0
ETF_pos = 0
defaultSize = 600
Thre = [5000,1000,3000,-1000,-3000]
safeValue = [2000,2000]
orderThre = [6,12]


N = 0
UB_UPPER = 60
UB_LOWER = 40
GEM_UPPER = 30
GEM_LOWER = 20
ETF_UPPER = 90
ETF_LOWER = 60

#Fllowed traderID
targetTrader=' '



def get_tick(session):
    resp = session.get('http://localhost:9999/v1/case')
    if resp.ok:
        case = resp.json()
        return case['tick']
    raise ApiException('Authorization error.Check api key.')
```

```
def position(session,ticker):
    payload={'ticker':ticker}
    resp = session.get('http://localhost:9999/v1/securities',
params=payload)
    if resp.ok:
        pos = resp.json()
        return pos[0]['position']
    raise ApiException('Authorization error.check api key')

def bidInfo(session, ticker):
    payload={'ticker':ticker}
    resp=session.get('http://localhost:9999/v1/securities/book',
params=payload)
    if resp.ok:
        book=resp.json()
        priceList=[]
        quantityList=[]
        for i in range(0, len(book['bids'])):
            priceList.append(book['bids'][i]['price'])
            quantityList.append(book['bids'][i]['quantity'])
        return priceList, quantityList


def askInfo(session, ticker):
    payload={'ticker':ticker}
    resp=session.get('http://localhost:9999/v1/securities/book',
params=payload)
    if resp.ok:
        book=resp.json()
        priceList=[]
        quantityList=[]
        for i in range(0, len(book['asks'])):
            priceList.append(book['asks'][i]['price'])
            quantityList.append(book['asks'][i]['quantity'])
        return priceList, quantityList


def news(session):
    resp = session.get('http://localhost:9999/v1/news')
    if resp.ok:
        n = resp.json()
        return n[0]['news_id'], n[0]['tick'], n[0]['body'], n[0]
['headline']
    raise ApiException('Authorization error.check api key')

def traderInfo(session, ticker):
    payload={'ticker':ticker}
    resp = session.get('http://localhost:9999/v1/securities/book',
params=payload)
    if resp.ok:
        book = resp.json()
        bidList=[]
        askList=[]
        for i in range(0, len(book['bids'])):
```

```
            bidList.append(book['bids'][i]['trader_id'])
        for j in range(0, len(book['asks'])):
            askList.append(book['asks'][j]['trader_id'])


        return bidList, askList
    raise ApiException('Authorization error.check api key')


def bound(s):
    news_id, news_tick, news_body, headline = news(s)
    global UB_UPPER
    global UB_LOWER
    global GEM_LOWER
    global GEM_UPPER
    global ETF_LOWER
    global ETF_UPPER
    global N
    if (news_id != N):
        result = news_body.split('$')[1:][0]
        estimate = float(result)
        up = round((299 - news_tick) / 50 + estimate,2)
        low = round(estimate - (299 - news_tick) / 50,2)

        if (headline[-3:] == ' UB'):
            UB_UPPER = min(UB_UPPER, up)
            UB_LOWER = max(UB_LOWER, low)

        else:
            GEM_UPPER = min(GEM_UPPER, up)
            GEM_LOWER = max(GEM_LOWER, low)

        N = N + 1
        ETF_UPPER = round(UB_UPPER + GEM_UPPER,2)
        ETF_LOWER = round(UB_LOWER + GEM_LOWER,2)









# (1) Price Discovery Stragety:
def Mispricing(session, UB_bid, UB_ask, GEM_bid, GEM_ask, ETF_bid,
ETF_ask,UB_pos,ETF_pos,GEM_pos):

    tempList=[]



    ubShortD=max(0, round(UB_bid-(UB_UPPER + M_SPREAD),2))
```

```
    tempList.append(ubShortD)

    ubLongD=max(0, round(UB_LOWER - M_SPREAD-UB_ask,2))
    tempList.append(ubLongD)

    gemShortD=max(0, round(GEM_bid-(GEM_UPPER + M_SPREAD),2))
    tempList.append(gemShortD)

    gemLongD=max(0, round(GEM_LOWER - M_SPREAD-GEM_ask,2))
    tempList.append(gemLongD)

    etfShortD=max(0, round(ETF_bid-(ETF_UPPER + M_SPREAD),2))
    tempList.append(etfShortD)

    etfLongD=max(0, round(ETF_LOWER - M_SPREAD-ETF_ask,2))
    tempList.append(etfLongD)

    tempList.sort()


    """only deal with the largest difference,  do we need to
consider about the second?"""
    if tempList[5]==ubShortD and ubShortD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':(UB_bid-
RISK1),'action':'SELL'})
        sleep(SPEEDBUMP)
    elif tempList[5]==gemShortD and gemShortD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':(GEM_bid-
RISK1),'action':'SELL'})
        sleep(SPEEDBUMP)
    elif tempList[5]==etfShortD and etfShortD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':(ETF_bid-
RISK1),'action':'SELL'})
        sleep(SPEEDBUMP)
    elif tempList[5]==ubLongD and ubLongD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':
(UB_ask+RISK1),'action':'BUY'})
        sleep(SPEEDBUMP)
    elif tempList[5]==gemLongD and gemLongD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':
(GEM_ask+RISK1),'action':'BUY'})
        sleep(SPEEDBUMP)
    elif tempList[5]==etfLongD and etfLongD>0:
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':
(ETF_ask+RISK1),'action':'BUY'})
        sleep(SPEEDBUMP)
```

```
    if (abs(UB_pos)+abs(ETF_pos)+abs(GEM_pos))>95000:
        si=100000-(abs(UB_pos)+abs(ETF_pos)+abs(GEM_pos))
        if tempList[5]==ubShortD and ubShortD>0 :
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'UB','type':'MARKET','quantity':si,'action':'SELL'})
            sleep(SPEEDBUMP)
        elif tempList[5]==gemShortD and gemShortD>0:
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'GEM','type':'MARKET','quantity':si,'action':'SELL'})
            sleep(SPEEDBUMP)
        elif tempList[5]==etfShortD and etfShortD>0:
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'ETF','type':'MARKET','quantity':si,'action':'SELL'})
            sleep(SPEEDBUMP)
        elif tempList[5]==ubLongD and ubLongD>0:
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'UB','type':'MARKET','quantity':si,'action':'BUY'})
            sleep(SPEEDBUMP)
        elif tempList[5]==gemLongD and gemLongD>0:
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'GEM','type':'MARKET','quantity':si,'action':'BUY'})
            sleep(SPEEDBUMP)
        elif tempList[5]==etfLongD and etfLongD>0:
            session.post('http://localhost:9999/v1/orders', params
= {'ticker':'ETF','type':'MARKET','quantity':si,'action':'BUY'})
            sleep(SPEEDBUMP)
```

```
# (2) Arbitrage Stragety:

"""start arb"""
def Arbitrage1(session, UB_bid, UB_ask, GEM_bid, GEM_ask, ETF_bid,
ETF_ask, short_ETF_size, long_ETF_size):
    """short etf and long two stocks"""
    if ETF_bid > (UB_ask + GEM_ask + SPREAD):
       session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'MARKET','quantity':short_ETF_size,'action':'
SELL'})
```

```
       session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'MARKET','quantity':short_ETF_size,'action':'B
UY'})
       session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'MARKET','quantity':short_ETF_size,'action':'
BUY'})

    """LONG ETF AND SHORT TWO STOCKS"""
    if (UB_bid + GEM_bid) > (ETF_ask + SPREAD):

        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'MARKET','quantity':long_ETF_size,'action':'B
UY'})
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'MARKET','quantity':long_ETF_size,'action':'SE
LL'})
        session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'MARKET','quantity':long_ETF_size,'action':'S
ELL'})

    sleep(SPEEDBUMP)



"""end the arb and close out position"""
def Arbitrage2(session, UB_bid, UB_ask, GEM_bid, GEM_ask, ETF_bid,
ETF_ask, short_ETF_size, long_ETF_size, UB_pos, GEM_pos, ETF_pos):


        """close out position --> short etf"""
        if ETF_bid > (UB_ask + GEM_ask+ SPREAD) and ETF_pos>0:
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'MARKET','quantity':min(short_ETF_size,abs(ETF
_pos)),'action':'BUY'})
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'MARKET','quantity':min(short_ETF_size,abs(ET
F_pos)),'action':'BUY'})
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'MARKET','quantity':min(short_ETF_size,abs(ET
F_pos)),'action':'SELL'})



        """long etf"""

        if (UB_bid + GEM_bid) > (ETF_ask+ SPREAD) and ETF_pos<0:
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'MARKET','quantity':min(long_ETF_size,abs(ETF_
pos)),'action':'SELL'})
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'MARKET','quantity':min(long_ETF_size,abs(ETF
_pos)),'action':'SELL'})
           session.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'MARKET','quantity':min(long_ETF_size,abs(ETF
```

```
_pos)),'action':'BUY'})

        sleep(SPEEDBUMP)
```

```
# (3) Marketmaking Stragety:

def UBmktmaking(K, SPREAD, s, UB_bid, UB_ask, UB_bid_size,
UB_ask_size, UB_pos):
    obsBid = []
    obsAsk = []
    bid_ask_spread = UB_ask[0] - UB_bid[0]

    for i in range(1,5):
        obsBid.append(sum(UB_bid_size[0:i])/ sum(UB_ask_size[0:i]))
        obsAsk.append(1/obsBid[i-1])

    if bid_ask_spread >= SPREAD:
        if abs(UB_pos) < Thre[0]:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
0]-0.01,'action':'SELL'})

            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    else:
        if UB_pos < Thre[1]:
            if obsBid[2] < K and obsBid[3] > K:
```

```
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
3]+0.01,'action':'BUY'})

            elif obsBid[1] < K and obsBid[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
2]+0.01,'action':'BUY'})

            elif obsBid[0] < K and obsBid[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
1]+0.01,'action':'BUY'})

            elif obsBid[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params = {'ticker':'UB', 'type':'LIMIT', 'quantity':defaultSize,
'price':UB_bid[0]+0.01, 'action':'BUY'})

            sleep(SPEEDBUMP)
        elif UB_pos > Thre[1] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)
        elif UB_pos > Thre[2] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)

        if UB_pos > Thre[3]:
            if obsAsk[2] < K and obsAsk[3] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
3]-0.01,'action':'SELL'})

            elif obsAsk[1] < K and obsAsk[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
2]-0.01,'action':'SELL'})

            elif obsAsk[0] < K and obsAsk[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
```

```
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
1]-0.01,'action':'SELL'})

            elif obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_ask[
0]-0.01,'action':'SELL'})

            sleep(SPEEDBUMP)
        elif UB_pos < Thre[3] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
        elif UB_pos < Thre[4] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'UB','type':'LIMIT','quantity':defaultSize,'price':UB_bid[
0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    resp = s.get('http://localhost:9999/v1/orders')
    openOrderNo = 0
    if resp.ok:
        openOrder=resp.json()
        for i in range(len(openOrder)):
            if openOrder[i]['ticker'] == 'UB':
                openOrderNo += 1

    if abs(UB_pos) > safeValue[0] and  openOrderNo > orderThre[0] :
        cancel_params = {'all': 1, 'ticker': 'UB'}
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)
    elif abs(UB_pos) < safeValue[1] and  openOrderNo >
orderThre[1] :
        cancel_params = {'all': 1, 'ticker': 'UB'}
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)



def GEMmktmaking(K, SPREAD, s, GEM_bid, GEM_ask, GEM_bid_size,
GEM_ask_size, GEM_pos):
    obsBid = []
    obsAsk = []
    bid_ask_spread = GEM_ask[0] - GEM_bid[0]

    for i in range(1,5):
        obsBid.append(sum(GEM_bid_size[0:i])/
```

```
sum(GEM_ask_size[0:i]))
        obsAsk.append(1/obsBid[i-1])

    if bid_ask_spread >= SPREAD:
        if abs(GEM_pos) < Thre[0]:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[0]-0.01,'action':'SELL'})

            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    else:
        if GEM_pos < Thre[1]:
            if obsBid[2] < K and obsBid[3] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[3]+0.01,'action':'BUY'})

            elif obsBid[1] < K and obsBid[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[2]+0.01,'action':'BUY'})

            elif obsBid[0] < K and obsBid[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[1]+0.01,'action':'BUY'})

            elif obsBid[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
        elif GEM_pos > Thre[1] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)
        elif GEM_pos > Thre[2] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
```

```
k[0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)

        if GEM_pos > Thre[3]:
            if obsAsk[2] < K and obsAsk[3] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[3]-0.01,'action':'SELL'})

            elif obsAsk[1] < K and obsAsk[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[2]-0.01,'action':'SELL'})

            elif obsAsk[0] < K and obsAsk[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[1]-0.01,'action':'SELL'})

            elif obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_as
k[0]-0.01,'action':'SELL'})

            sleep(SPEEDBUMP)
        elif GEM_pos < Thre[3] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
        elif GEM_pos < Thre[4] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'GEM','type':'LIMIT','quantity':defaultSize,'price':GEM_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    resp = s.get('http://localhost:9999/v1/orders')
    openOrderNo = 0
    if resp.ok:
        openOrder=resp.json()
        for i in range(len(openOrder)):
            if openOrder[i]['ticker'] == 'GEM':
                openOrderNo += 1

    if abs(GEM_pos) > safeValue[0] and  openOrderNo > orderThre[0] :
        cancel_params = {'all': 1, 'ticker': 'GEM'}
```

```
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)
    elif abs(GEM_pos) < safeValue[1] and  openOrderNo >
orderThre[1] :
        cancel_params = {'all': 1, 'ticker': 'GEM'}
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)

def ETFmktmaking(K, SPREAD, s, ETF_bid, ETF_ask, ETF_bid_size,
ETF_ask_size, ETF_pos):
    obsBid = []
    obsAsk = []
    bid_ask_spread = ETF_ask[0] - ETF_bid[0]

    for i in range(1,5):
        obsBid.append(sum(ETF_bid_size[0:i])/
sum(ETF_ask_size[0:i]))
        obsAsk.append(1/obsBid[i-1])

    if bid_ask_spread >= SPREAD:
        if abs(ETF_pos) < Thre[0]:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[0]-0.01,'action':'SELL'})

            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    else:
        if ETF_pos < Thre[1]:
            if obsBid[2] < K and obsBid[3] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[3]+0.01,'action':'BUY'})

            elif obsBid[1] < K and obsBid[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[2]+0.01,'action':'BUY'})

            elif obsBid[0] < K and obsBid[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[1]+0.01,'action':'BUY'})
```

```
            elif obsBid[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
        elif ETF_pos > Thre[1] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)
        elif ETF_pos > Thre[2] and obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[0]-0.01,'action':'SELL'})

                sleep(SPEEDBUMP)

        if ETF_pos > Thre[3]:
            if obsAsk[2] < K and obsAsk[3] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[3]-0.01,'action':'SELL'})

            elif obsAsk[1] < K and obsAsk[2] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[2]-0.01,'action':'SELL'})

            elif obsAsk[0] < K and obsAsk[1] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[1]-0.01,'action':'SELL'})

            elif obsAsk[0] > K:
                resp = s.post('http://localhost:9999/v1/orders',
params =
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_as
k[0]-0.01,'action':'SELL'})

            sleep(SPEEDBUMP)
        elif ETF_pos < Thre[3] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[0]+0.01,'action':'BUY'})
```

```
            sleep(SPEEDBUMP)
        elif ETF_pos < Thre[4] and obsBid[0] > K:
            resp = s.post('http://localhost:9999/v1/orders', params
=
{'ticker':'ETF','type':'LIMIT','quantity':defaultSize,'price':ETF_bi
d[0]+0.01,'action':'BUY'})

            sleep(SPEEDBUMP)
    resp = s.get('http://localhost:9999/v1/orders')
    openOrderNo = 0
    if resp.ok:
        openOrder=resp.json()
        for i in range(len(openOrder)):
            if openOrder[i]['ticker'] == 'ETF':
                openOrderNo += 1

    if abs(ETF_pos) > safeValue[0] and  openOrderNo > orderThre[0] :
        cancel_params = {'all': 1, 'ticker': 'ETF'}
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)
    elif abs(ETF_pos) < safeValue[1] and  openOrderNo >
orderThre[1] :
        cancel_params = {'all': 1, 'ticker': 'ETF'}
        s.post('http://localhost:9999/v1/commands/cancel', params =
cancel_params)
        sleep(SPEEDBUMP)

def closeP(s, ticker, pos, ask, bid):
    a = abs(pos) / 5000
    b = abs(pos) % 5000
    while a/2 > 0 :
        if pos > 0:
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':ticker,'type':'LIMIT','quantity':5000,'price':ask,
'action':'SELL'})
        elif pos < 0:
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':ticker,'type':'LIMIT','quantity':5000,'price':bid,'action'
:'BUY'})
        sleep(SPEEDBUMP)
        a -= 1
    if pos > 0:
        s.post('http://localhost:9999/v1/orders', params =
{'ticker':ticker,'type':'LIMIT','quantity':b,'price':ask,'action':'S
ELL'})
    elif pos < 0:
        s.post('http://localhost:9999/v1/orders', params =
{'ticker':ticker,'type':'LIMIT','quantity':b,'price':bid,'action':'B
UY'})
    sleep(SPEEDBUMP)
```

```
# (4) Special feature:   insider trading
def insidertrading(s, UB_ask_size, GEM_ask_size,
ETF_ask_size,UB_bid_size, GEM_bid_size, ETF_bid_size, UB_bid,
GEM_bid, ETF_bid, UB_ask, GEM_ask, ETF_ask):
    #collect the trader info first
    bidTradersU=[]
    bidTradersG=[]
    bidTradersE=[]
    askTradersU=[]
    askTradersG=[]
    askTradersE=[]

    bidTradersU, askTradersU=traderInfo(s, 'UB')
    bidTradersG, askTradersG=traderInfo(s, 'GEM')
    bidTradersE, askTradersE=traderInfo(s, 'ETF')

    indexList = []
    if bidTradersU.count(targetTrader) > 0:
        for i in range(0, len(bidTradersU)):
            if bidTradersU[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':UB_bid_size[indexList[i]] ,
'price':UB_bid[indexList[i]]+RISK1,'action':'BUY'})
        sleep(SPEEDBUMP)

    indexList = []
    if bidTradersG.count(targetTrader) > 0:
        for i in range(0,len(bidTradersG)):
            if bidTradersG[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':GEM_bid_size[indexList[i]]
,'price':GEM_bid[indexList[i]]+RISK1,'action':'BUY'})
        sleep(SPEEDBUMP)

    indexList = []
    if bidTradersE.count(targetTrader) > 0:
        for i in range(0,len(bidTradersE)):
            if bidTradersE[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':ETF_bid_size[indexList[i]]
```

```
,'price':ETF_bid[indexList[i]]+RISK1,'action':'BUY'})
        sleep(SPEEDBUMP)

    indexList = []
    if askTradersU.count(targetTrader) > 0:
        for i in range(0,len(askTradersU)):
            if askTradersU[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':UB_ask_size[indexList[i]] ,
'price':UB_ask[indexList[i]]-RISK1,'action':'SELL'})
        sleep(SPEEDBUMP)

    indexList = []
    if askTradersG.count(targetTrader) > 0:
        for i in range(0,len(askTradersG)):
            if askTradersG[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':GEM_ask_size[indexList[i]]
,'price':GEM_ask[indexList[i]]-RISK1,'action':'SELL'})
        sleep(SPEEDBUMP)

    indexList = []
    if askTradersE.count(targetTrader) > 0:
        for i in range(0,len(askTradersE)):
            if askTradersE[i] == targetTrader:
                indexList.append(i)
        for i in range(0,len(indexList)):
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':ETF_ask_size[indexList[i]]
,'price':ETF_ask[indexList[i]]-RISK1,'action':'SELL'})
        sleep(SPEEDBUMP)
```

```
# (5) Special feature:  Freelunch 1

def freelunch1(s,tick):
```

```
    if tick==10:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-5,'ac
tion':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+5,'ac
tion':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-5,'
action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+5,'
action':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-5,'
action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+5,'
action':'SELL'})
    elif tick==50:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-8,'ac
tion':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+8,'ac
tion':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-8,'
action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+8,'
action':'SELL'})
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-8,'
action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+8,'
action':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':40,'action':'B
```

```
UY'})
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':60,'action':'S
ELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':20,'action':'
BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':30,'action':'
SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':60,'action':'
BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':90,'action':'
SELL'})

    elif tick==100:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-10,'a
ction':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+10,'a
ction':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+10,
'action':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+10,
'action':'SELL'})
    elif tick==150:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-10,'a
ction':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+10,'a
```

```
ction':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+10,
'action':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+10,
'action':'SELL'})
    elif tick==200:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-10,'a
ction':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+10,'a
ction':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+10,
'action':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-10,
'action':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+10,
'action':'SELL'})
    elif tick==250:

            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_LOWER-10,'a
ction':'BUY'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_UPPER+10,'a
ction':'SELL'})
            sleep(BUMP)
            s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_LOWER-10,
```

```
'action':'BUY'})
           sleep(BUMP)
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_UPPER+10,
'action':'SELL'})
           sleep(BUMP)
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_LOWER-10,
'action':'BUY'})
           sleep(BUMP)
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_UPPER+10,
'action':'SELL'})
```

```
# (6) Special feature:  Freelunch 2

def
freelunch2(s,ubl,ual,gbl,gal,ebl,eal,UB_bid,UB_ask,GEM_bid,GEM_ask,E
TF_bid,ETF_ask):
      if len(UB_bid)<ubl/2 and len(UB_bid)<20 and
UB_bid[0]<UB_LOWER :
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_bid[0]-5,'a
ction':'BUY'})
      if len(UB_ask)<ual/2 and len(UB_ask)<20 and
UB_ask[0]>UB_UPPER:
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'UB','type':'LIMIT','quantity':5000,'price':UB_ask[0]+5,'a
ction':'SELL'})

      if len(GEM_bid)<gbl/2 and len(GEM_bid)<20 and
GEM_bid[0]<GEM_LOWER:
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_bid[0]-10
,'action':'BUY'})

      if len(GEM_ask)<gal/2 and len(GEM_ask)<20 and
GEM_ask[0]>GEM_UPPER:
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'GEM','type':'LIMIT','quantity':5000,'price':GEM_ask[0]+10
,'action':'SELL'})

      if len(ETF_bid)<ebl/2 and len(ETF_bid)<20 and
ETF_bid[0]<ETF_LOWER :
           s.post('http://localhost:9999/v1/orders', params =
```

```
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_bid[0]-15
,'action':'BUY'})

      if len(ETF_ask)<eal/2 and len(ETF_ask)<20 and
ETF_ask[0]>ETF_UPPER:
           s.post('http://localhost:9999/v1/orders', params =
{'ticker':'ETF','type':'LIMIT','quantity':5000,'price':ETF_ask[0]+15
,'action':'SELL'})
      sleep(SPEEDBUMP)
```

```
def main():
    with requests.Session() as s:
        s.headers.update(API_KEY)
        tick = get_tick(s)

        UB_bid = []
        UB_ask = []
        GEM_bid = []
        GEM_ask = []
        ETF_bid = []
        ETF_ask = []
        UB_bid_size = []
        UB_ask_size = []
        GEM_bid_size = []
        GEM_ask_size = []
        ETF_bid_size = []
        ETF_ask_size = []
        tt=0
        ubl=0
        ual=0
        gbl=0
        gal=0
        ebl=0
        eal=0
        while tick > 0 and tick < 300 and not shutdown:



            UB_bid, UB_bid_size=bidInfo(s,'UB')
            GEM_bid, GEM_bid_size=bidInfo(s,'GEM')
```

```
            ETF_bid, ETF_bid_size=bidInfo(s,'ETF')

            UB_ask, UB_ask_size=askInfo(s,'UB')
            GEM_ask, GEM_ask_size=askInfo(s,'GEM')
            ETF_ask, ETF_ask_size=askInfo(s,'ETF')

            UB_pos = position(s,'UB')
            GEM_pos = position(s,'GEM')
            ETF_pos = position(s,'ETF')




            bound(s)

            # (1)Price Discovery Stragety
            Mispricing(s,UB_bid[0], UB_ask[0], GEM_bid[0],
GEM_ask[0], ETF_bid[0], ETF_ask[0],UB_pos,ETF_pos,GEM_pos)



            # (2)Arbitrage Stragety
            '''
            short_ETF_size = min(ETF_bid_size[0]/5, UB_ask_size[0]/
5, GEM_ask_size[0]/5, 1000)
            long_ETF_size = min(ETF_ask_size[0]/5, UB_bid_size[0]/5,
GEM_bid_size[0]/5, 1000)
            if tick < 40 :
               Arbitrage1(s, UB_bid[0], UB_ask[0], GEM_bid[0],
GEM_ask[0], ETF_bid[0], ETF_ask[0], short_ETF_size, long_ETF_size)

            elif N==0:

               Arbitrage2(s, UB_bid[0], UB_ask[0], GEM_bid[0],
GEM_ask[0], ETF_bid[0], ETF_ask[0], short_ETF_size, long_ETF_size,
UB_pos, GEM_pos, ETF_pos)
            '''

            # (3)Marketmaking Stragety
            '''
            if tick < 40 :
                UBmktmaking(1, SPREAD, s, UB_bid, UB_ask,
UB_bid_size, UB_ask_size, UB_pos)
                GEMmktmaking(1, SPREAD, s, GEM_bid, GEM_ask,
GEM_bid_size, GEM_ask_size, GEM_pos)
                ETFmktmaking(1, SPREAD, s, ETF_bid, ETF_ask,
ETF_bid_size, ETF_ask_size, ETF_pos)
            elif tick < 50:
                cancel_params = {'all': 1}
                s.post('http://localhost:9999/v1/commands/cancel',
params = cancel_params)

                closeP(s,'UB',UB_pos,UB_bid[0], UB_ask[0])
                closeP(s,'GEM',GEM_pos,GEM_bid[0], GEM_ask[0])
```

```
                    closeP(s,'ETF',ETF_pos,ETF_bid[0], ETF_ask[0])
                '''

                # (4) Special feature:  insider trading
                insidertrading(s, UB_ask_size, GEM_ask_size,
ETF_ask_size,UB_bid_size, GEM_bid_size, ETF_bid_size, UB_bid,
GEM_bid, ETF_bid, UB_ask, GEM_ask, ETF_ask)



                # (5) Special feature:  Freelunch 1
                freelunch1(s,tick)


                # (6) Special feature:  Freelunch 2
                ''' when there is dramatical change of the number of the
orders in the book:
                if tick>tt:

freelunch2(s,ubl,ual,gbl,gal,ebl,eal,UB_bid,UB_ask,GEM_bid,GEM_ask,E
TF_bid,ETF_ask)
                   ubl=len(UB_bid)
                   ual=len(UB_ask)
                   gbl=len(GEM_bid)
                   gal=len(GEM_ask)
                   ebl=len(ETF_bid)
                   eal=len(ETF_ask)
                   tt=tick
                '''

                tick=get_tick(s)


if __name__ == '__main__':
    signal.signal(signal.SIGINT, signal_handler)
    main()
```