In [ ]:

```
# https://www.cmegroup.com/market-data/files/cme-term-sofr-reference-rates-benchmark-methodology.pdf
# https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm
```

In [2]:

```
import datetime as dt
import numpy as np
```

In [3]:

```
# define sofr fwd rate curve
today = dt.date(2023, 2, 1)
fomc_dates = [dt.date(2023, 3, 22), dt.date(2023, 5, 3), dt.date(2023, 6, 14)]
days = (dt.date(2023, 2, 5) - today).days
fwd_times = [(fomc_dates[i] - today).days/365 for i in range(3)]
fwd_times.insert(0, 0.0)
fwd_values = np.zeros(shape=4)
```

In [4]:

```
# function to get daily forward rate at time t
def get_daily_forward_rate(t, fwd_times, fwd_values):
  fwd = 0.0
  # code here
  return fwd
```

In [6]:

```
# compute futures price implied from fwd rate curve
# - need to define futures contract: start time and no. of days in contract period
# example: 1M sofr future
fut_start_dates = [dt.date(2023, i, 1) for i in range(2, 7)]
fut_start_times = [(fut_start_dates[i] - today).days/365 for i in range(5)]
fut_period_lengths = [(dt.date(2023, i+1, 1) - dt.date(2023, i, 1)).days for i in range(2, 7)]
```

In [7]:

```
def implied_price(start_time, period_length, fwd_times, fwd_values):
  price = 0.0
  # code here
  return price
```

In [8]:

```
# define pricing error function
# x = ndarray of jump sizes
def price_error(x, fut_mkt_prices, fut_start_times, fut_period_lengths, fwd_times, fwd_values):
  error = 0.0
  # code here
  return error
```

In [9]:

```
from scipy import optimize
```

In [ ]:

```
jump_sizes = np.zeros(shape=4)
optimize.minimize(price_error, x0=jump_sizes, args=(fut_start_times, fut_period_lengths, fwd_times, fwd_values),
                  method='BFGS')
```