Hw3 Q1

Formulation:-

$$Ls[i] = \begin{cases} 0 & \text{when } i = 0 \\ L[i] + \max(no\text{-}job[i-1], Ls[i-1], hs[i-1]) & \text{otherwise} \end{cases}$$

$$hs[i] = \begin{cases} 0 & \text{when } i = 0 \\ h[i] + \max(Ls[i-2], nojob[i-1]) & \text{otherwise} \end{cases}$$

$\max(Ls, hs)$

Algorithm:

```
def optimalvalue(n, l, h):
    ljobs = [0]
    hjobs = [0]
    ljobs[0] = L[0]
    hjobs[0] = h[0]
    For i in range(n):
        ljobs = max(ljobs[i-1] + l[i], l[i] + hjobs[i-1])
        hjobs = max(h[i] + ljobs[i-2], h[i] + hjobs[i-2])

    return max(ljobs[-1], hjobs[-1])
```

HW 3Q2

(a)

i	1	2	3	4	5
x_i	2	3	4	1	5
f(i)	1	2	3	4	5

The given algorithm activate EMP in 4th second, result in Total of 3 + 1 + 2, while activating EMP on 3rd and 5th results in 3 + 2 = 5

(b) Formulation:-

$$OPT[i] = \max(OPT[i-j] + \min(f[j], x[i]))$$

Algorithm

```
def scheduleEmp(n, f, x)
OPT = []
For i in range(n):
    maxOPT = 0
    For j in rang(2, i):
        if OPT[i-j] + min(f[j], x[i]) > maxOPT:
            maxOPT = OPT[i-j] + min(f[j], x[i])

    OPT[i] = maxOPT

return OPT[n]
```

Hw 3Q3

Algorithm

- Construct a graph with a node for each stock and a directed edge (i, j) for each pair of stocks.

  cost of each edge is $-\log(r_{ij})$

- a trading cycle is an opportunity cycle if and only if:

  $\prod r_{ij} > 1$: for all (i, j) in the trading cycle.

- Use Bellman-ford algorithm to detect to whether an opportunity cycle exits.

  As bellman-ford is used to detect negative cycles. Also we know, cycle is opportunity cycle if its the negative cycle also.

Hw 3Q4

Formulation:

$$M[i][j] = \begin{cases} 0 & \text{if } i = j \\ \min\limits_{k=1}^{j-1} \left\{ M[i][k] + M[k+1][j] + p_{i-1} \cdot c_k \cdot c_j \right. \end{cases}$$

Algorithm:-

```
def MatrixOrdering(x):
    m = [][]
    For i in range(2, len(x)):
        For j in range(1, len(x) - i + 1):
            Q = j + i - 1
            For k in rang(i, Q):
                c = m[j][k] + m[k+1][Q] + p[j-1] * p[k] * p[Q]
                if c < m[j][Q]
                    m[j][Q] = c

    return m[1][n]
```