①

$$Opt(i) = \begin{cases} \emptyset & i = \emptyset \\ \max(\ell_1, h_1) & i = 1 \\ \max\{\ell_i + opt(i-1), \\ \quad h_i + Opt(i-2)\} \end{cases}$$

(a)

(b)

$M[0] = \emptyset$

$M[1] = Max(\ell_1, h_1)$

for $i = 2$ to n

$\quad M[i] = \max(\ell_i + M[i-1],$

$\quad\quad h_i + M[i-2])$

---

③

(a) The Same example provided in the question

1-st and 4-th Seconds

1-st: 1 robot, 4-th: 1 robot

↳ 2 robots total ; $\boxed{opt = 5}$

(b)

$$opt(i) = \begin{cases} 0 & i = \emptyset \\ \max\{\min(x_i, f(i)), \\ \quad \max_{\forall j < i}\{opt(j) + \\ \quad\quad \min(x_i, f(i-j)\}\} \end{cases}$$

(2-c)

$M[\emptyset] = \emptyset$

for $i = 1$ to n

$\quad M[i] = \min(X_i, f(i))$

$\quad$ for $j = 1$ to $i-1$

$\quad\quad M[i] = \max\{M[i],$

$\quad\quad\quad \min(X_i, f(i-j)) + M[j]\}$

---

3) (10pts)

① Mapping to neg-cycle discovery

Consider a cycle $r_{i_1 i_2} \cdots r_{i_{k-1} i_k}$

The cycle is beneficial if

$$\prod_{1 \le j < k} r_{i_j i_{j+1}} > 1$$

$$\Rightarrow \prod_{1 \le j < k} \frac{1}{r_{i_j i_{j+1}}} < 1$$

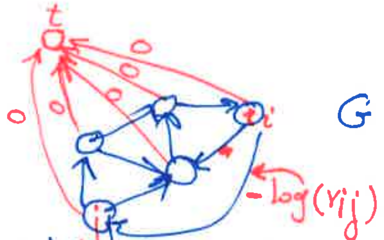
Taking $\boxed{\log}$ from the two Sides

$$\sum_{1 \le j < k} \log\left(\frac{1}{r_{i_j i_{j+1}}}\right) < \emptyset$$

$$\Rightarrow \boxed{\sum -\log(r_{i_j i_{j+1}}) < \emptyset}$$

⇒ we Can replace the edge weights with $-\log(r_{ij})$ and find Neg-cycles

3-B (10pts)




```
Detect Neg-Cycle (G(V∪{t}, E))
    foreach node v ∈ V
        M[v] = ∞
    M[t] = 0
    for i = 1 to n {
        foreach node w ∈ V {
            if (M[w] has been updated in
                Prev. step)
            {
                for each node v  s.t. (v,w) ∈ E
                {
                    if (M[v] > M[w] + C_vw)
                    {
                        M[v] = M[w] + C_vw
                        Successor[v] = w
                        if (i == n)
                            return Cycle(Successor, v)
                    }
                }
            }
        }
    return None // not found
}
```

3b - Continued

```
Cycle (Successor, v)
{
    tmp = v ; C = [v]
    Repeat {
        tmp = tmp.Successor
        C.add(tmp)
    until (tmp = v)
    return C
}
```

4) (a)

$$Opt(i,j) = \begin{cases} 0 & i=j \\ \min\limits_{i \le k < j} \left(opt(i,k) + opt(k+1,j) + A[i-1]A[k]A[j] \right) & \end{cases}$$

15 pts →

(b)

```
Matrix-Mul(X)
    n = |X| - 1
    for i = 1 to n
        M[i,i] = 0
    for l = 2 to n
        for i = 1 to n-l+1
            j = i+l-1
            M[i,j] = ∞
            for k = i to j-1
                q = M[i,k] + M[k+1,j] +
                    A[i-1] A[k] A[j]
                if (q < M[i,j])
                    M[i,j] = q
                    S[i,j] = k
    return M, S
```