Advanced Algorithms Analysis and Design

By

Nazir Ahmad Zafar

Lecture No 17

Assembly-Line Scheduling Problem

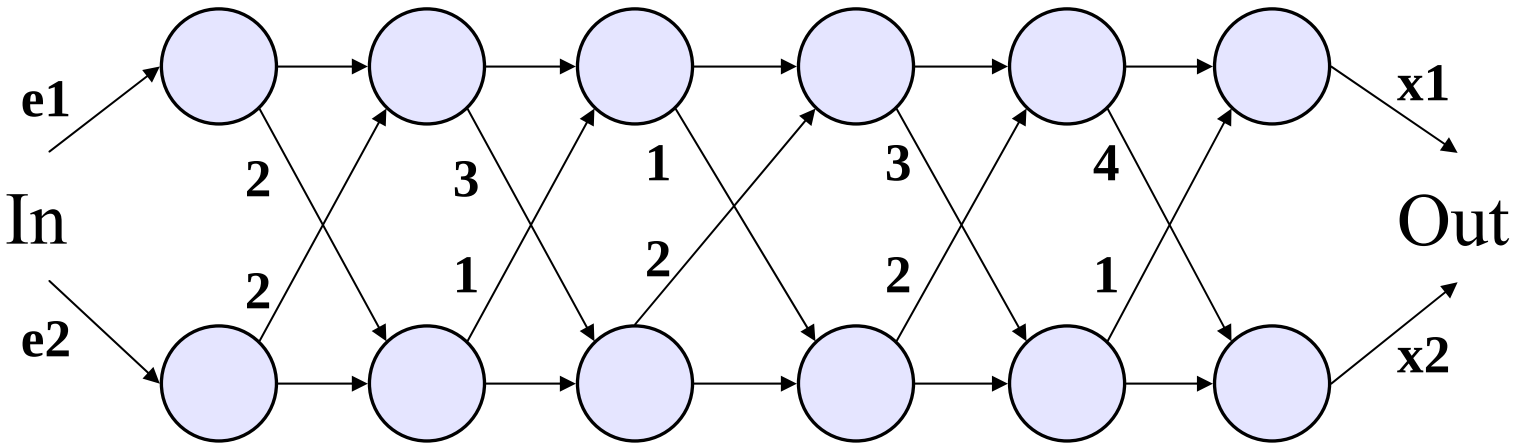
Today Covered

- Assembly Line Scheduling Problem
- Problem Analysis
  - Defining Notations
  - Brute Force approach
  - Dynamic Solution
- Algorithm using Dynamic Programming
- Time Complexity
- Generalization and Applications
- Conclusion

Assembly-Line Scheduling Problem

- There are two assembly lines each with n stations
- The jth station on line i is denoted by $S_{i,\, j}$
- The assembly time at that station is $a_{i,j}$.
- An auto enters factory, goes into line i taking time e_i
- After going through the jth station on a line i, the auto goes on to the (j+1)st station on either line
- There is no transfer cost if it stays on the same line
- It takes time $t_{i,j}$ to transfer to other line after station $S_{i,j}$
- After exiting the nth station on a line, it takes time x_i for the completed auto to exit the factory.
- Problem is to determine which stations to choose from lines 1 and 2 to minimize total time through the factory.

Notations: Assembly-Line Scheduling Problem



Stations $S_{i,j}$;

2 assembly lines, $i = 1,2$;

n stations, $j = 1,\ldots,n$.

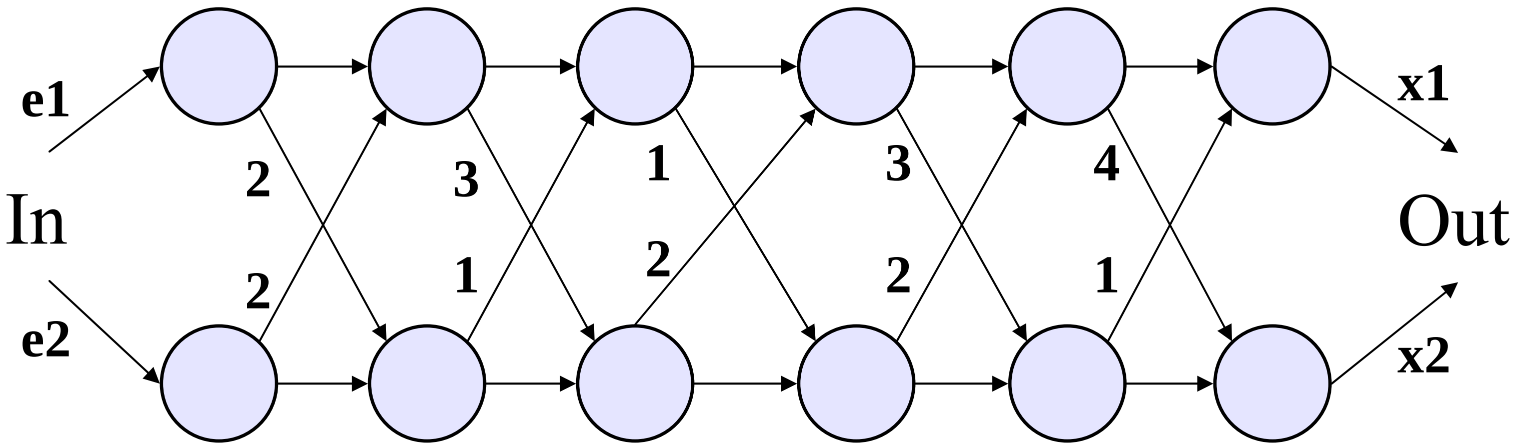
$a_{i,j}$ = assembly time at $S_{i,j}$;

$t_{i,j}$ = transfer time from $S_{i,j}$ (to $S_{i-1,j+1}$ OR $S_{i+1,j+1}$);

e_i = entry time from line i;

x_i = exit time from line i .

Brute Force Solution



Total Computational Time

= possible ways to enter in stations at level n x one way Cost

Possible ways to enter in stations at level 1 = 2^1

Possible ways to enter in stations at level 2 = $2^2 \ldots$

Possible ways to enter in stations at level 2 = 2^n

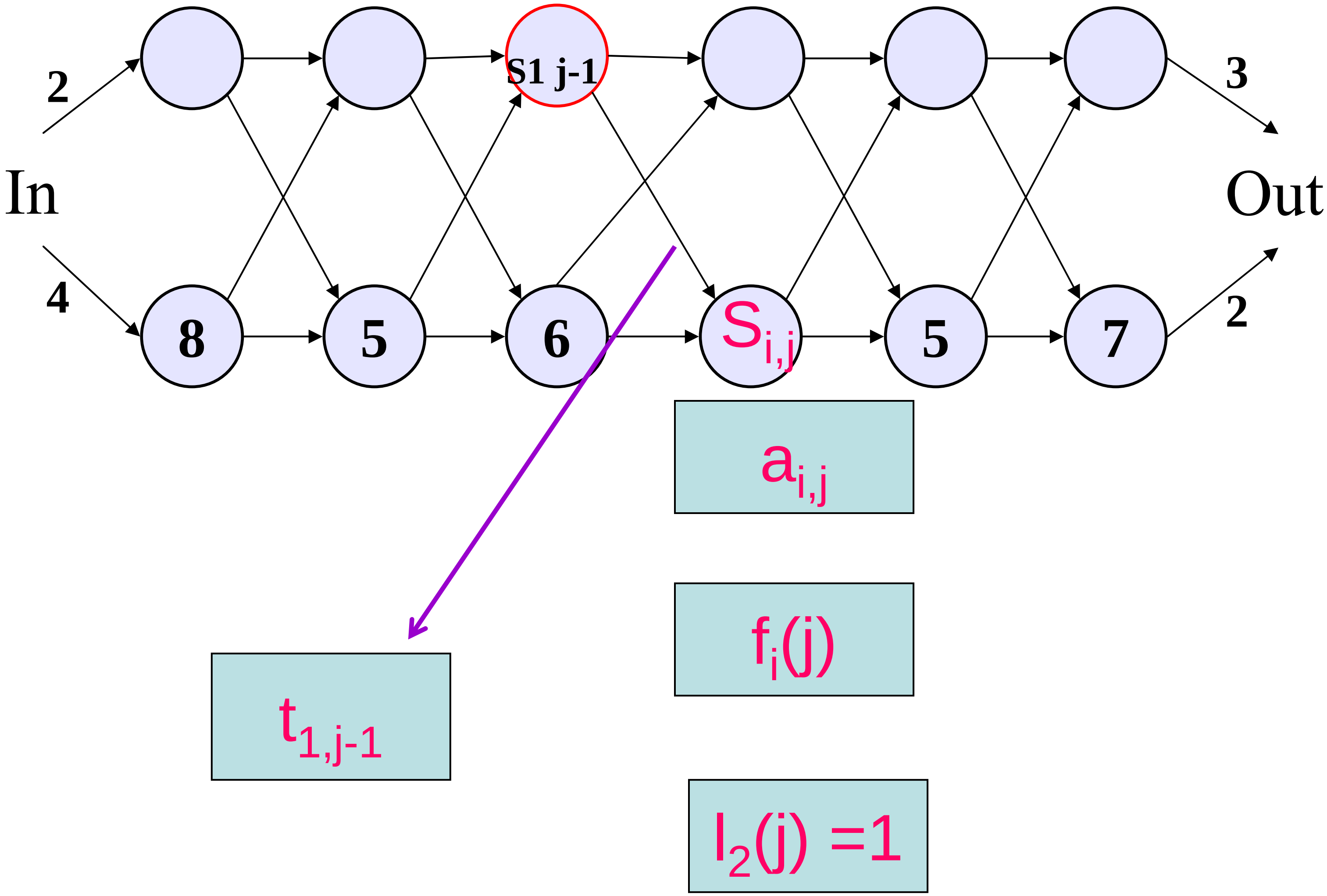
Total Computational Time = $n.2^n$

Dynamic Programming Solution

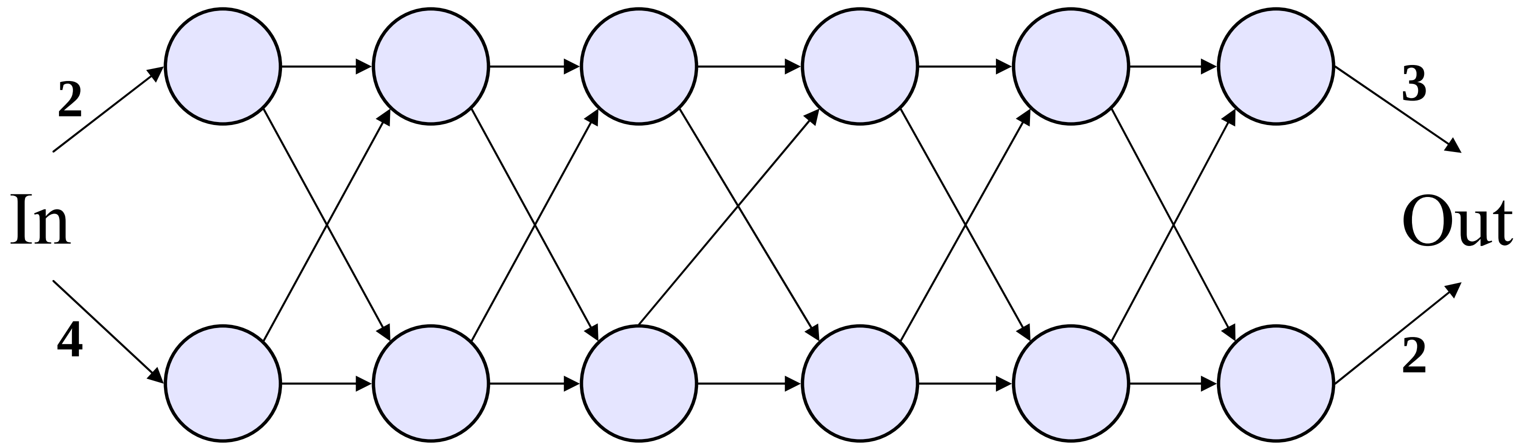
Notations: Finding Objective Function

- Let $f_i[j]$ = fastest time from starting point station $S_{i,\ j}$
- $f_1[n]$ = fastest time from starting point station $S_{1\ n}$
- $f_2[n]$ = fastest time from starting point station $S_{2\ n}$
- $l_i[j]$ = The line number, 1 or 2, whose station j-1 is used in a fastest way through station $S_{i,\ j}$.
- It is to be noted that $l_i[1]$ is not required to be defined because there is no station before 1
- $t_i[j-1]$ = transfer time from line i to station $S_{i-1,\ j}$ or $S_{i+1,\ j}$
- Objective function = $f^* = \min(f_1[n] + x_1,\ f_2[n] + x_2)$
- l^* = to be the line no. whose n^{th} station is used in a fastest way.

Notations: Finding Objective Function



Mathematical Model: Finding Objective Function



$f_1[1] = e_1 + a_{1,1};$

$f_2[1] = e_2 + a_{2,1}.$

$f_1[j] = \min\ (f_1[j-1] + a_{1,j},\ f_2[j-1] + t_{2,j-1} + a_{1,j})$ for $j \geq 2;$

$f_2[j] = \min\ (f_2[j-1] + a_{2,j},\ f_1[j-1] + t_{1,j-1} + a_{2,j})$ for $j \geq 2;$

Complete Model: Finding Objective Function

Base Cases

- $f_1[1] = e_1 + a_{1,1}$
- $f_2[1] = e_2 + a_{2,1}$

Two possible ways of computing $f_1[j]$

- $f_1[j] = f_2[j-1] + t_{2, j-1} + a_{1, j}$ OR $f_1[j] = f_1[j-1] + a_{1, j}$

  For $j = 2, 3, \ldots, n$

  $$f_1[j] = \min (f_1[j-1] + a_{1, j},\ f_2[j-1] + t_{2, j-1} + a_{1, j})$$

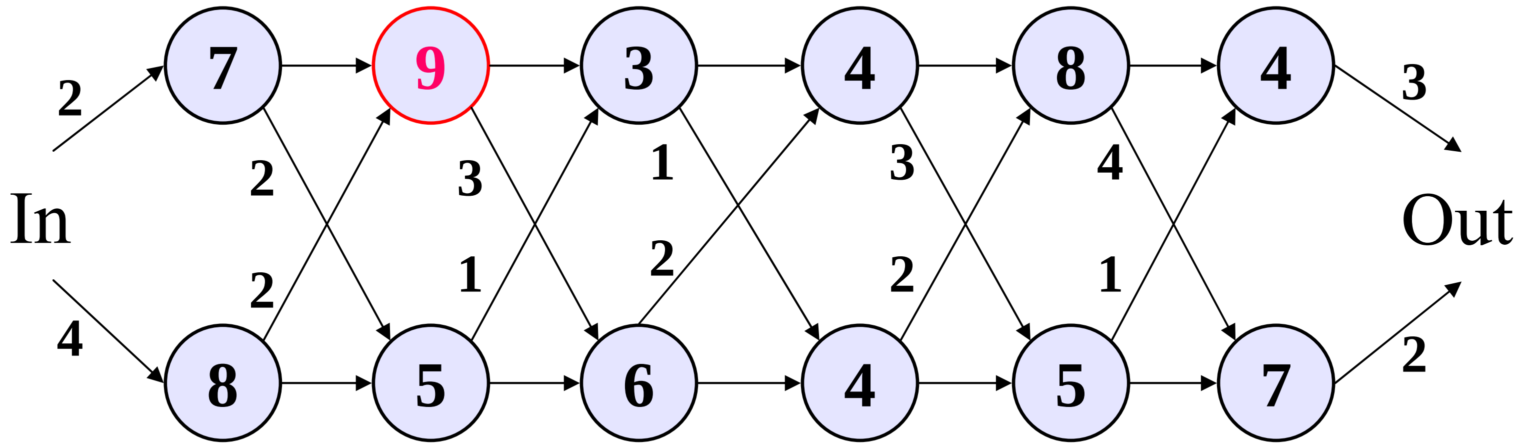
Symmetrically

For $j = 2, 3, \ldots, n$

$$f_2[j] = \min (f_2[j-1] + a_{2, j},\ f_1[j-1] + t_{1, j-1} + a_{2, j})$$

Objective function $= f^* = \min(f_1[n] + x_1,\ f_2[n] + x_2)$

Example: Computation of f1[2]



- $f_1[1] = e_1 + a_{1,1} = 2 + 7 = 9$
- $f_2[1] = e_2 + a_{2,1} = 4 + 8 = 12$

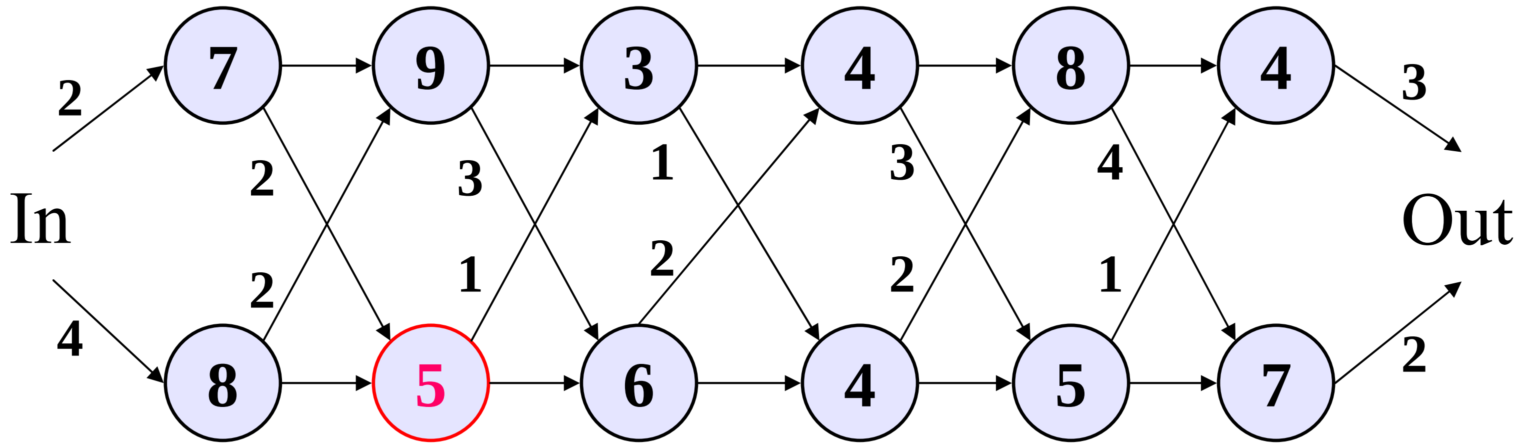
$f_1[j] = \min (f_1[j-1] + a_{1,\ j},\ f_2[j-1] + t_{2,\ \ j-1} + a_{1,\ j})$

$f_2[j] = \min (f_2[j-1] + a_{2,\ j},\ f_1[j-1] + t_{1,\ j-1} + a_{2,\ j})$

$j = 2$

$f_1[2] = \min (f_1[1] + a_{1,\ 2},\ f_2[1] + t_{2,\ \ 1} + a_{1,\ 2})$

$= \min (9 + 9,\ 12 + 2 + 9) = \min (18,\ 23) = 18,\ l_1[2] = 1$



- $f_1[1] = e_1 + a_{1,1} = 2 + 7 = 9$
- $f_2[1] = e_2 + a_{2,1} = 4 + 8 = 12$

$f_1[j] = \min (f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j})$

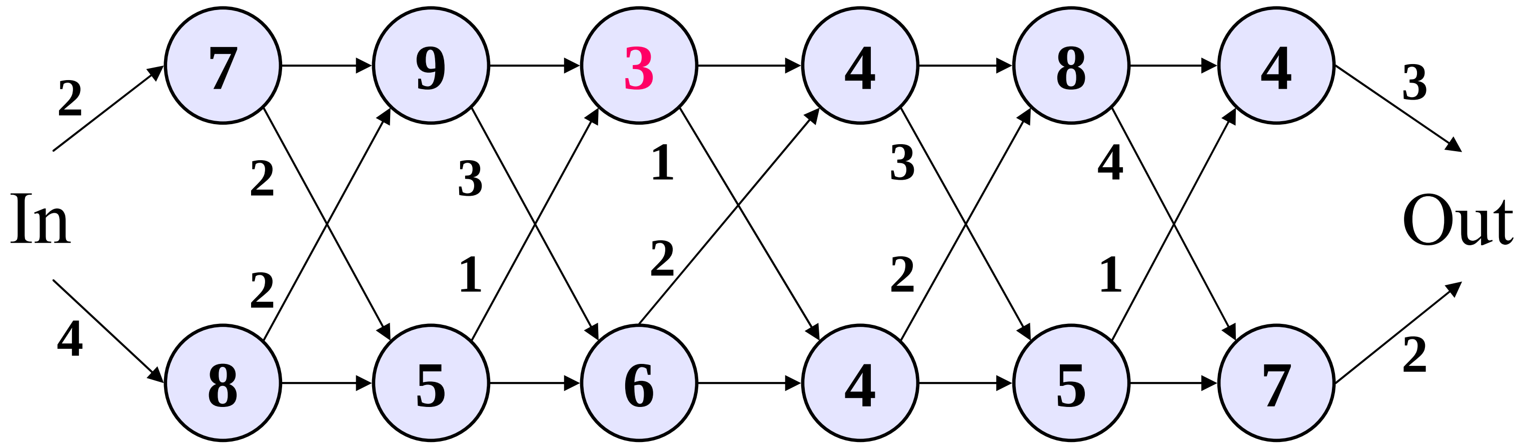
$f_2[j] = \min (f_2[j-1] + a_{2,j}, f_1[j-1] + t_{1,j-1} + a_{2,j})$

$j = 2$

$f_2[2] = \min (f_2[1] + a_{2,2}, f_1[1] + t_{1,1} + a_{2,2})$

$= \min (12 + 5, 9 + 2 + 5) = \min (17, 16) = 16,$ $l_2[2] = 1$

Computation of f1[3]



$f_1[j] = \min(f_1[j-1] + a_{1,j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,j})$

$f_2[j] = \min(f_2[j-1] + a_{2,j},\ f_1[j-1] + t_{1,j-1} + a_{2,j})$

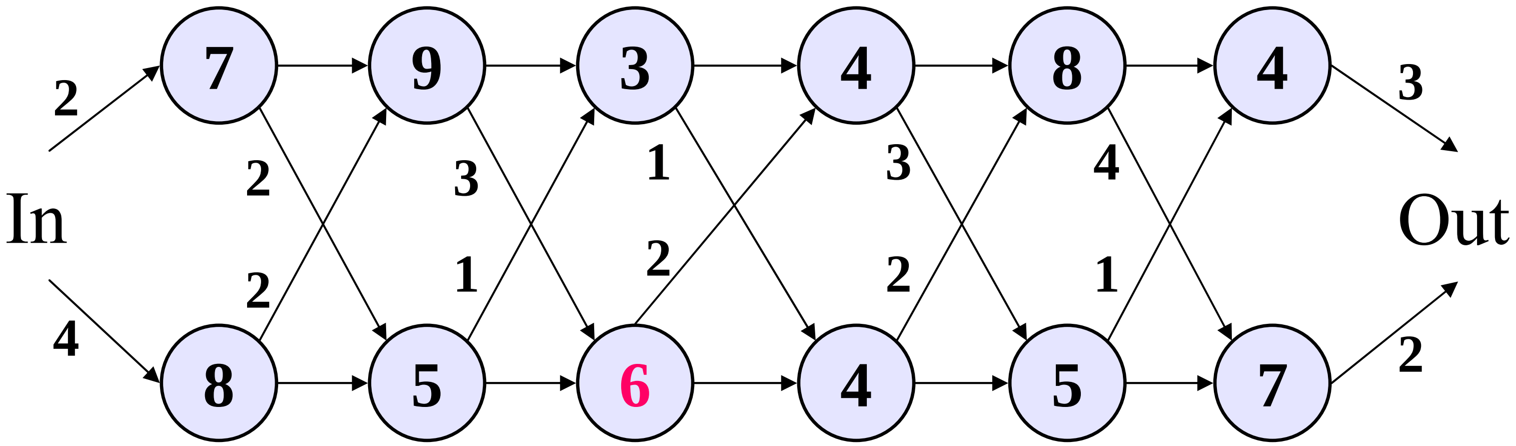
$j = 3$

$f_1[3] = \min(f_1[2] + a_{1,3},\ f_2[2] + t_{2,\ 2} + a_{1,3})$

$= \min(18 + 3,\ 16 + 1 + 3)$

$= \min(21, 20) = 20,$ $l_1[3] = 2$

Computation of f2[3]



$f_1[j] = \min(f_1[j-1] + a_{1,j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,j})$

$f_2[j] = \min(f_2[j-1] + a_{2,j},\ f_1[j-1] + t_{1,j-1} + a_{2,j})$

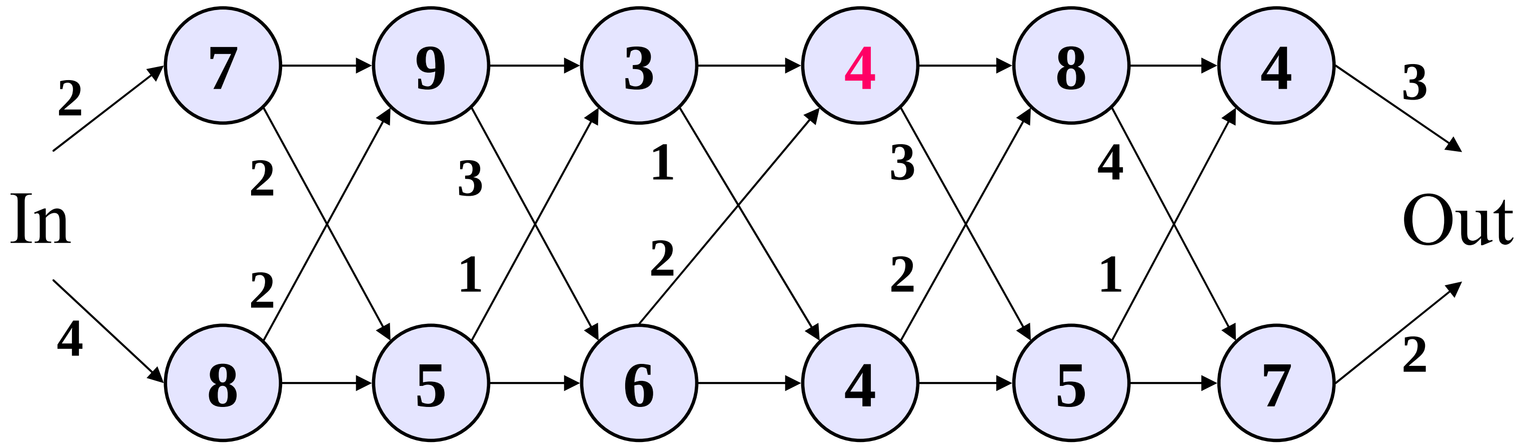
$j = 3$

$f_2[3] = \min(f_2[2] + a_{2,3},\ f_1[2] + t_{1,2} + a_{2,3})$

$= \min(16 + 6,\ 18 + 3 + 6)$

$= \min(22, 27) = 22,$ $\quad l_2[3] = 2$

Computation of f1[4]

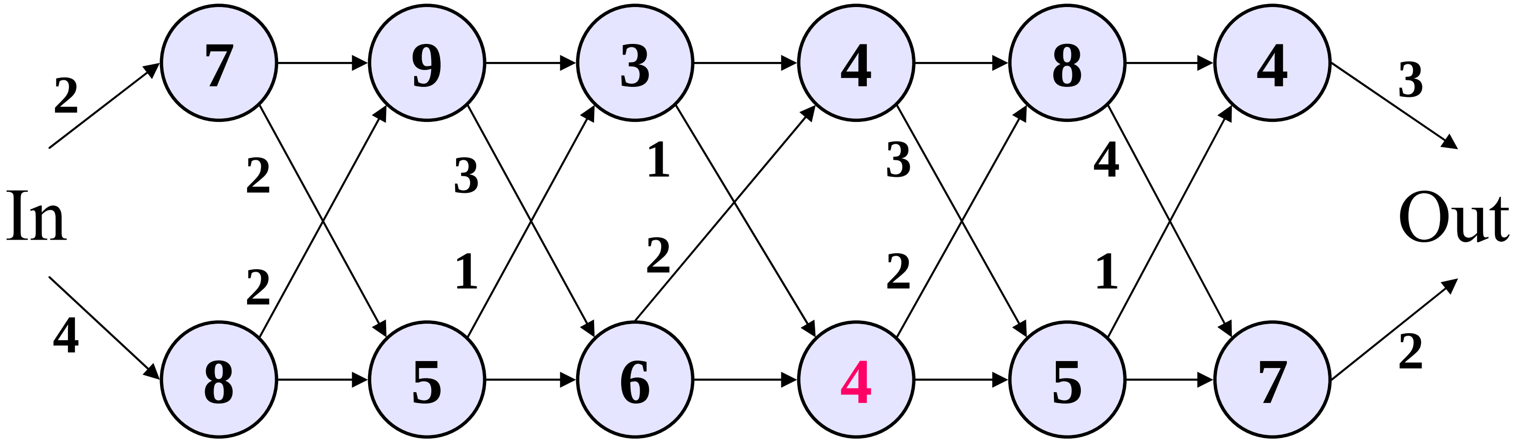


$f_1[j] = \min(f_1[j-1] + a_{1,j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,j})$

$f_2[j] = \min(f_2[j-1] + a_{2,j},\ f_1[j-1] + t_{1,j-1} + a_{2,j})$

$j = 4$

$$f_1[4] = \min(f_1[3] + a_{1,4},\ f_2[3] + t_{2,3} + a_{1,4})$$
$$= \min(20 + 4,\ 22 + 1 + 4)$$
$$= \min(24, 27) = 24, \qquad l_1[4] = 1$$



$f_1[j] = \min(f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j})$

$f_2[j] = \min(f_2[j-1] + a_{2,j}, f_1[j-1] + t_{1,j-1} + a_{2,j})$

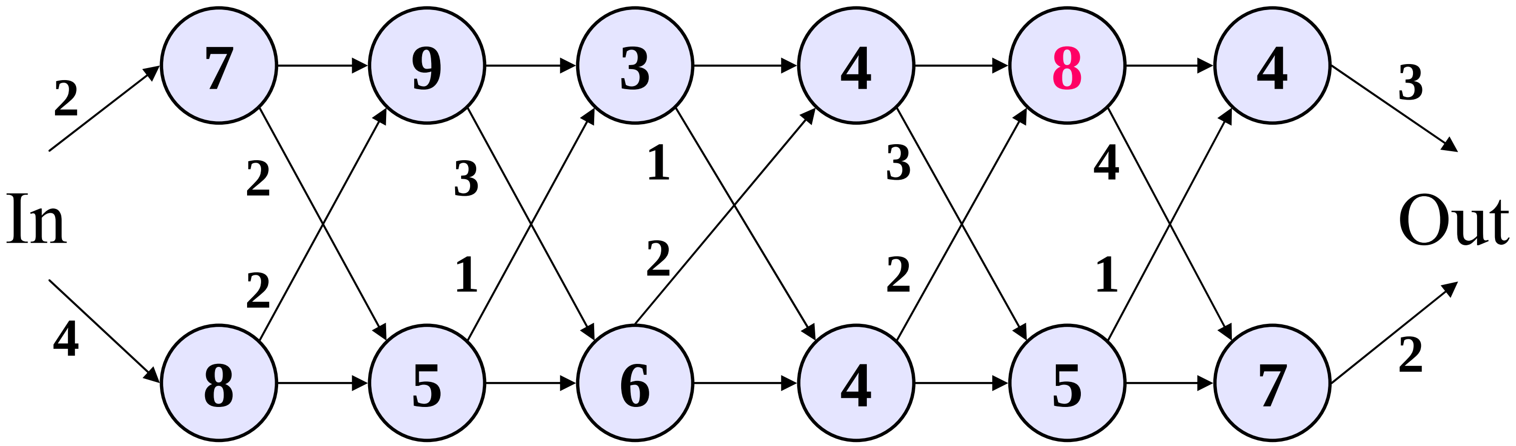
$j = 4$

$f_2[4] = \min(f_2[3] + a_{2,4}, f_1[3] + t_{1,3} + a_{2,4})$

$= \min(22 + 4, 20 + 1 + 4)$

$= \min(26, 25) = 25,$ $\quad l_2[4] = 1$

Computation of f1[5]



$$f_1[j] = \min (f_1[j-1] + a_{1,\,j},\ f_2[j-1] + t_{2,\,j-1} + a_{1,\,j})$$

$$f_2[j] = \min (f_2[j-1] + a_{2,\,j},\ f_1[j-1] + t_{1,\,j-1} + a_{2,\,j})$$

$j = 5$

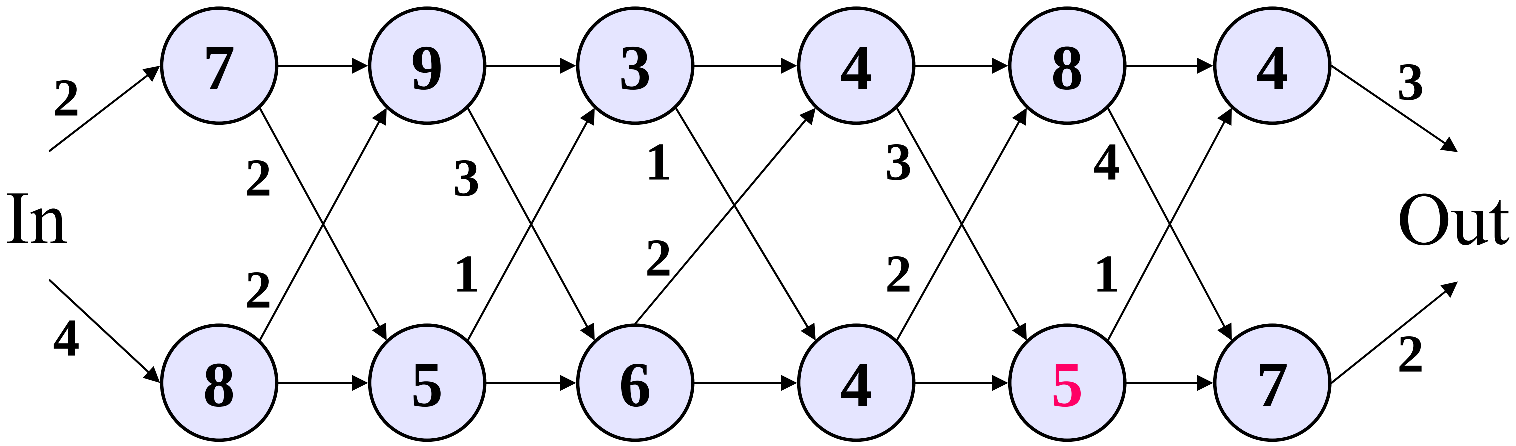
$$f_1[5] = \min (f_1[4] + a_{1,\,5},\ f_2[4] + t_{2,\,4} + a_{1,\,5})$$

$$= \min (24 + 8,\ 25 + 2 + 8)$$

$$= \min (32, 35) = 32,$$

$l_1[5] = 1$

Computation of f2[5]



$f_1[j] = \min(f_1[j-1] + a_{1,j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,j})$

$f_2[j] = \min(f_2[j-1] + a_{2,j},\ f_1[j-1] + t_{1,j-1} + a_{2,j})$

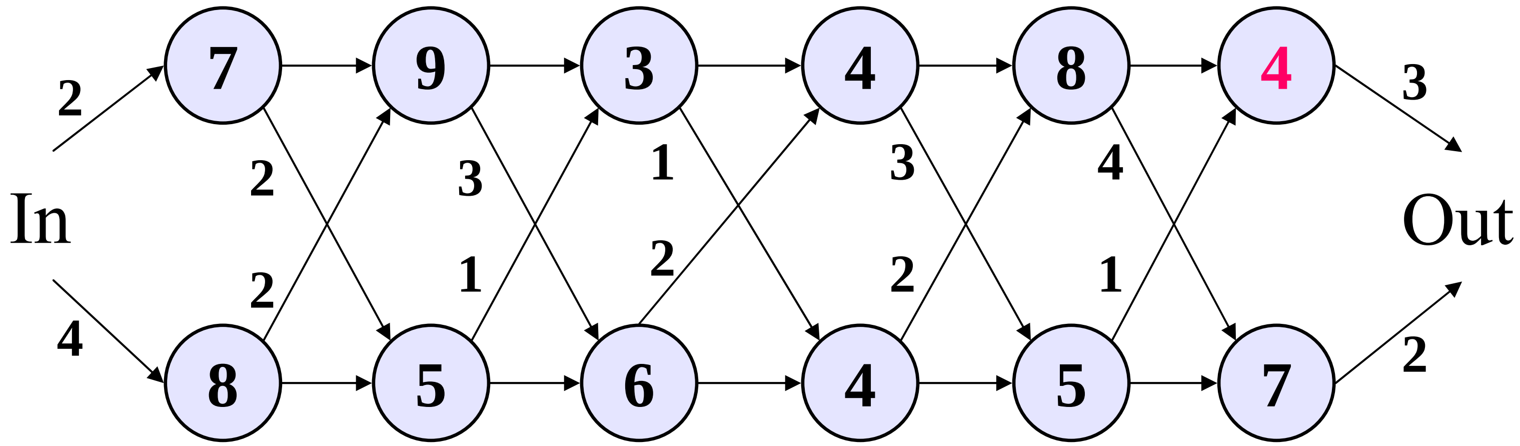
$j = 5$

$f_2[5] = \min(f_2[4] + a_{2,5},\ f_1[4] + t_{1,4} + a_{2,5})$

$= \min(25 + 5,\ 24 + 3 + 5)$

$= \min(30, 32) = 30,$ $\qquad l_2[5] = 2$

Computation of f1[6]



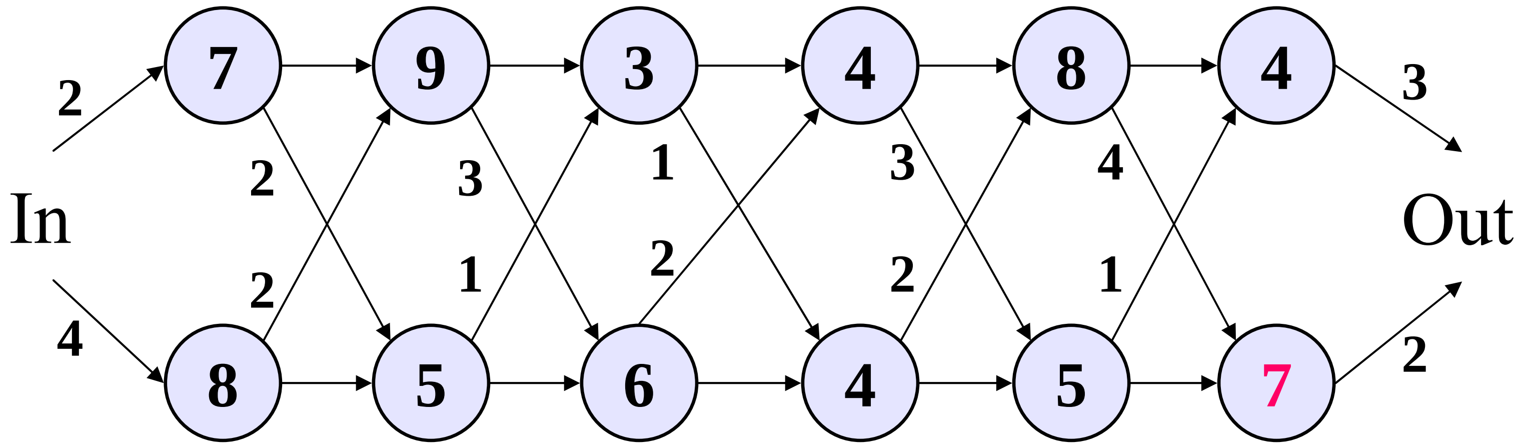
$$f_1[j] = \min\left(f_1[j-1] + a_{1,\,j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,\,j}\right)$$
$$f_2[j] = \min\left(f_2[j-1] + a_{2,\,j},\ f_1[j-1] + t_{1,\,j-1} + a_{2,\,j}\right)$$

$j = 6$

$$f_1[6] = \min\left(f_1[5] + a_{1,\,6},\ f_2[5] + t_{2,\,5} + a_{1,\,6}\right)$$
$$= \min(32 + 4,\ 30 + 1 + 4)$$
$$= \min(36,\ 35) = 35,$$

$l_1[6] = 2$

Computation of f2[6]



$f_1[j] = \min (f_1[j-1] + a_{1,\ j},\ f_2[j-1] + t_{2,\ j-1} + a_{1,\ j})$

$f_2[j] = \min (f_2[j-1] + a_{2,\ j},\ f_1[j-1] + t_{1,\ j-1} + a_{2,\ j})$

$j = 6$

$f_2[6] = \min (f_2[5] + a_{2,\ 6},\ f_1[5] + t_{1,\ 5} + a_{2,\ 6})$

$= \min (30 + 7,\ 32 + 4 + 7)$

$= \min (37,\ 43) = 37,$ $\quad l_2[6] = 2$

Keeping Track Constructing Optimal Solution

$$f^* = \min(f_1[6] + x_1, f_2[6] + x_2)$$

$$= \min(35 + 3, 37 + 2)$$

$$= \min(38, 39) = 38$$

$l^* = 1$

$l^* = 1 \Rightarrow$ Station $S_{1,6}$

$l_1[6] = 2 \Rightarrow$ Station $S_{2,5}$

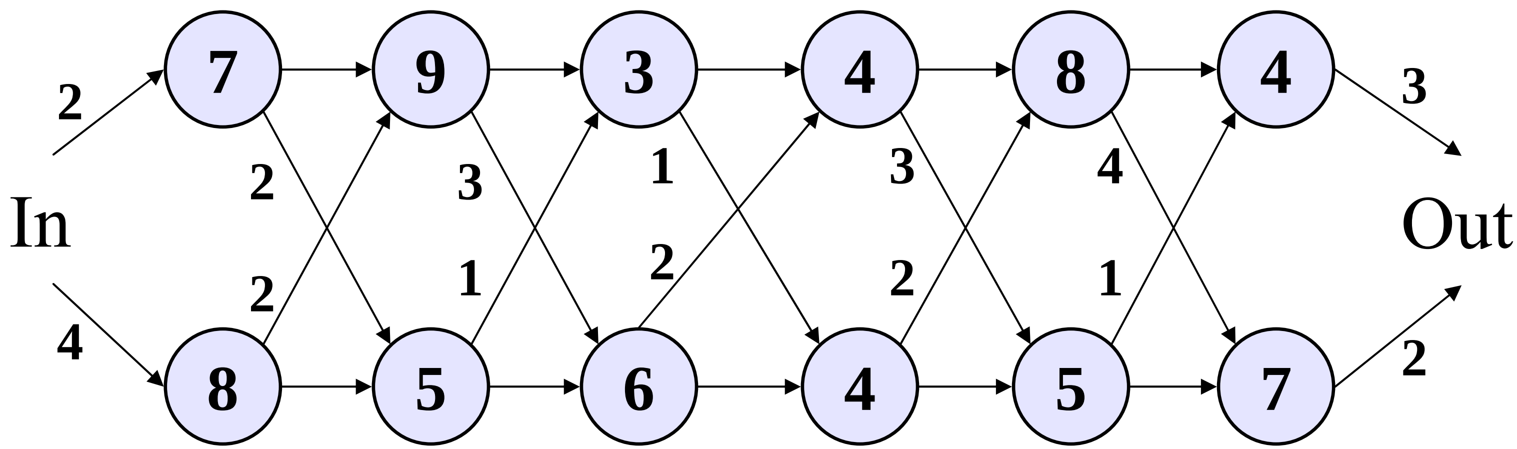
$l_2[5] = 2 \Rightarrow$ Station $S_{2,4}$

$l_2[4] = 1 \Rightarrow$ Station $S_{1,3}$

$l_1[3] = 2 \Rightarrow$ Station $S_{2,2}$

$l_2[2] = 1 \Rightarrow$ Station $S_{1,1}$

Entire Solution Set: Assembly-Line Scheduling



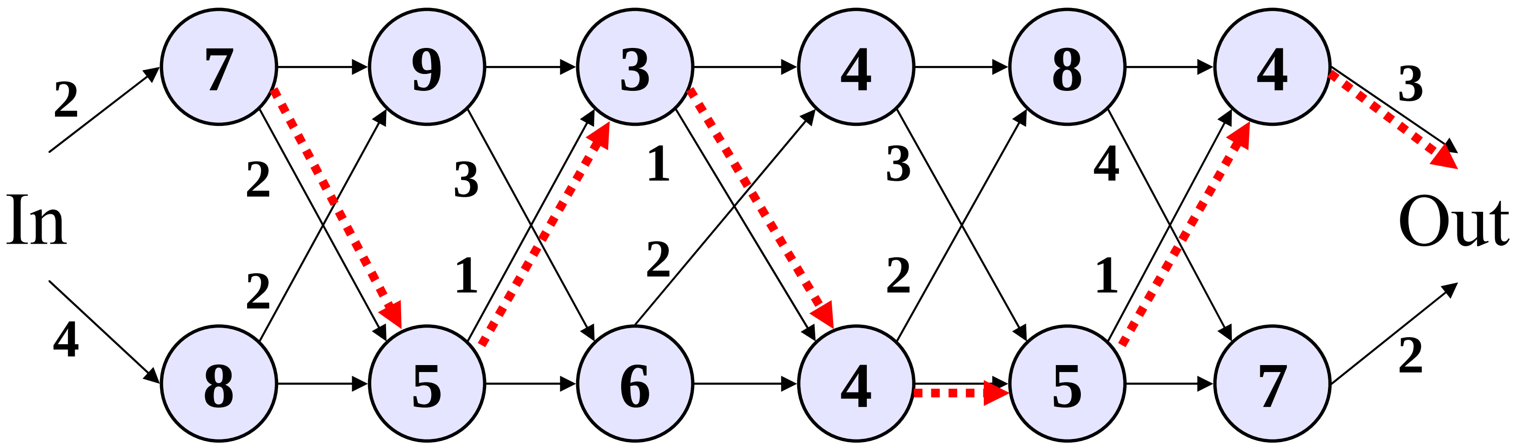
j / $f_i(j)$	1	2	3	4	5	6
1	9	18	20	24	32	35
2	12	16	22	25	30	37

$f^* = 38$

j / $l_i(j)$	2	3	4	5	6
1	1	2	1	1	2
2	1	2	1	2	2

$l^* = 1$

Fastest Way: Assembly-Line Scheduling



l^* = 1 => Station $S_{1,\ 6}$

$l_1[6]$ = 2 => Station $S_{2,\ 5}$

$l_2[5]$ = 2 => Station $S_{2,\ 4}$

$l_2[4]$ = 1 => Station $S_{1,\ 3}$

$l_1[3]$ = 2 => Station $S_{2,\ 2}$

$l_2[2]$ = 1 => Station $S_{1,\ 1}$