Sequencing problems:

Suppose there are finite number of different service facilities available and a finite number of jobs are to be processed through these service facilities in some pre-assigned order. The sequencing problem is to determine that order (sequence) of jobs for which the total cost (time) is minimum. Suppose an industry produces 'n' products (jobs) & let each of these are to be processed through m-different machines (service facilities). The order in which the products are to be processed through m-machines is specified. One is interested in finding that sequence in which the jobs should be processed through m-machines to optimize the total elapsed time. There are $(L^n)^m$ different sequences the jobs can be processed. This is a very large number. Here is a method to solve such problems.

Let there be n jobs to be performed on m-different machines.

M_{ij} → denotes the processing time required by i^{th} job on the j^{th} machine $(i = 1, 2, \cdots, n, \; j = 1, 2, \cdots, m)$.

T_{ij} → denotes the idle time on machine j from the completion of $(i-1)^{th}$ job to the start of i^{th} job.

The problem is to minimize the total elapsed time.

Assumptions:

- No machine can process more than one job at a time.
- Each job, once started on a machine, must be completed before the start of next job.
- Processing time M_{ij}'s are independent of the order of processing the jobs.
- The time required to transfer a job from one machine

to another is negligible.

- All jobs are known and are ready to start processing before the period under consideration begins.
- There is only one of each type of machine.

Algorithm: "n-jobs & two-machines" — order of Machines: M_1, M_2 — Seq. M_1, M_2

1. Select the smallest processing time among M_{i1}'s and M_{i2}'s. (i.e. $\min_i \{M_{i1}, M_{i2}\}$)

2. a) If this smallest processing time is for M_{i1} say M_{r1} then process r^{th} job first.

   b) If this smallest processing time is for M_{i2}, say M_{s2} then process s^{th} job at the last.

2. In case of tie,

   (a) If the tie is among M_{i1}'s and M_{i2}'s say $M_{r1} = M_{s2}$ then process r^{th} job first & s^{th} job at the last.

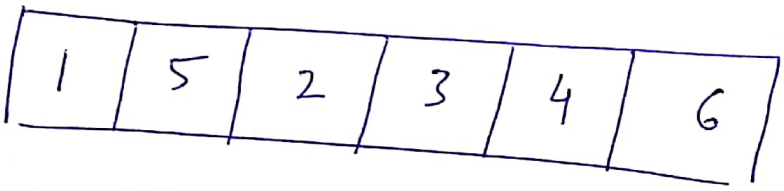
   (b). If the tie is among M_{i1}'s say $M_{p1} = M_{q1}$ then first of all process job corresponding to which M_{i2} is smallest.

   (c). If the tie is among M_{i2}'s say $M_{n2} = M_{m2}$ then process ~~the~~ in the last the job corresp. to which M_{i1} is smallest.

3. Cross the jobs assigned by step 1 & step 2. Repeat the above process till all the jobs are assigned.

Example 2. There are six jobs each of which has to be processed through two machines M_1 and M_2 in the order M_1, M_2. The processing times in hours are given below. Determine the sequence of these jobs which minimizes the total elapsed time:

Job	1	2	3	4	5	6
machine M_1	4	10	16	10	12	9
" M_2	8	9	8	6	12	2



Job	machine M_1		machine M_2		Idle time for M_2
	Time in	Time out	Time in	Time out	
1	0	4	4	12	4
5	4	16	16	28	4
2	16	26	28	37	0
3	26	42	42	50	5
4	42	52	52	58	2
6	52	61	61	63	3
					18

Idle time for machine M_2 = 18 Hrs.

" " " " M_1 = 63 − 61 = 2 Hrs.

Total elapsed time = 63 Hrs.

(3)

Ex 2. 5 - jobs. 2 - machines order M_1, M_2

Job	1	2	3	4	5
M_1	4	5	4	9	6
M_2	5	4	1	7	4

1	4	5	2	3

form the table.

Idle time for machine M_2 = 8 Hrs.

" " M_1 = 29 - 28 = 01 Hrs.

Total elapsed time = 29 Hrs.

Explaination see next page.

Example - 2

Seq.	Machines	M_1	M_2	Idle time for M_2
1	—	0–4	4–9	4
4	—	4–13	13–20	13–9 = 4
5	—	13–19	20–24	0
2	—	19–24	24–28	0
3	—	24–28	28–29	0
			↑ Total elapsed time	8 hrs. ↑ Idle time for M_2

Total elapsed time to complete all the jobs = 29 Hrs.

Idle Time for machine M_1 = (Total completion time) − (Total time taken by machine M_1)

= 29 − 28 = 01 Hrs.

Problem with n-jobs & m-machines:

Let there be n-jobs to be processed through m-machines $M_1, M_2, \ldots, M_m$ in the order $M_1, M_2 \ldots M_m$. If either or both of the following conditions hold then this problem can be reduced to the problem with n-jobs and two machines.

The conditions are:

$$\min_i M_{i1} \geq \max_i M_{ij} \quad \text{for } j = 2, 3, \ldots m-1$$

& or

$$\min_i M_{im} \geq \max_i M_{ij}, \quad \text{for } j = 2, 3, \ldots, m-1.$$

If neither of the above cond'ns hold, then there is no general procedure available.

Suppose either or both the cond'ns hold, then the procedure is as follows:

Step 1. Let there be two fictitious machines M_{F1} & M_{F2}. Calculate

$$M_{iF1} = M_{i1} + M_{i2} + \cdots + M_{i(m-1)}, \quad i = 1, 2, \ldots, n$$

$$M_{iF2} = M_{i2} + M_{i3} + \cdots + M_{im}, \quad i = 1, 2, \ldots, n.$$

where M_{iF1} & M_{iF2} $(i = 1, 2, \ldots n)$ are the processing times for the job i on the machines M_{F1} and M_{F2}, resp.

Step 2. Find optimal sequence for n-jobs with two machines M_{F1} & M_{F2} in the order M_{F1}, M_{F2}. This optimal sequence is also an optimal seq. for n-jobs & m-machines.

Ex. Five jobs, 3-machines, order M_1, M_2, M_3.

Job	1	2	3	4	5	
M_1	8	5	4	6	5	] F1
M_2	6	2	9	7	4	] F2
M_3	10	13	11	10	12	

$\min_i M_{i1} = 4$, $\min_i M_{i3} = 10$, $\max_i M_{i2} = 9$

$\therefore \min_i M_{i3} \geq \max_i M_{i2}$ (condition holds).

Therefore: the above problem reduces to:

Job	1	2	3	4	5
M_{F1}	14	7	13	13	9
M_{F2}	16	15	20	17	16

optimal sequence:

2	5	4	3	1

I → in ; O → out

Job	Machine M_1 I	M_1 O	M_2 I	M_2 O	M_3 I	M_3 O	Idle time (M_2)
2	0	5	5	7	7	20	7
5	5	10	10	14	20	32	0
4	10	16	16	23	32	42	0
3	16	20	23	32	42	53	0
1	20	28	32	38	53	63	0
							07

Idle time for M_3 : 07 Hrs.

" " M_2 : $5+3+2+0+0+(63-38) = 10 \text{ Hrs} + 25 = \underline{35 \text{ Hrs}}$.

" " M_1 : $63 - 28 = 35$ Hrs.

Total elapsed time = 63 Hrs.

Ex 2.

	1	2	3	4	5
M_1	7	10	8	9	7
M_2	2	1	4	0	5
M_3	5	6	3	7	2
M_4	8	7	12	10	9

Total elapsed time = 60 hr

ANSWER

TRY - YOURSELF

Processing two jobs through m-machines

Let there be two jobs A and B each of which is to be processed on m-machines say $M_1, M_2, \ldots, M_m$ in two different orders. [If order is same, then it will be 2-jobs and m-machines problem, for which earlier techniques can be used to solve]. The technological ordering of each of the two jobs through m-machines is known in advance. Such ordering may not be same for both the jobs. The exact or expected processing times on the given machines are known. Each machine can perform only one job at a time. The objective is to determine an optimal sequence of processing the jobs so as to minimize total elapsed time.

For solving, we use graphical approach. The procedure can be illustrated by taking examples.

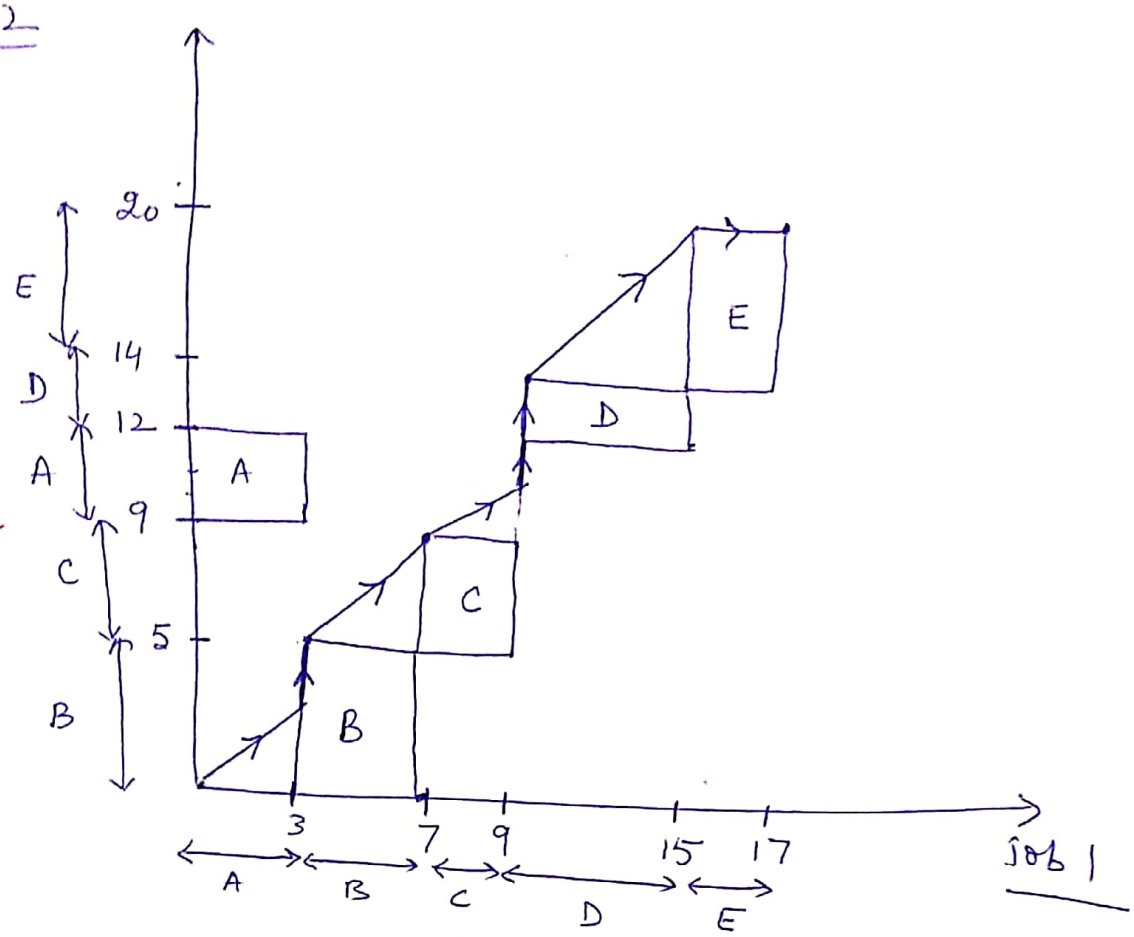
Ex 1. Use the graphical method to determine the minimum time needed to process the following jobs on the machines shown, i.e. each machine find the job which should be done first. Also, calculate the total elapsed time to complete both jobs.

		machine				
Job 1	Sequence	A	B	C	D	E
	Time (hrs)	3	4	2	6	2

		machine				
Job 2	Sequence	B	C	A	D	E
	Time (hrs)	5	4	3	2	6

Solution

Job 2



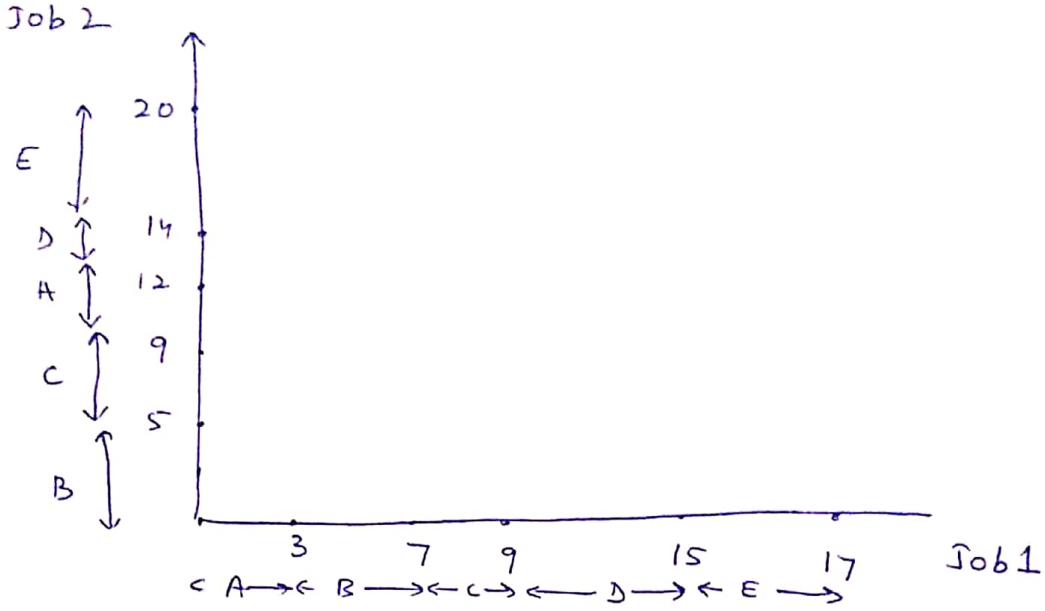
(i) Explaination (see next pages)

- If we move along x-axis, it represents processing time for Job 1 while Job 2 is idle. Similarly, moving along y-axis means processing time for Job 2 while Job 1 is idle.
- Moving horizontally means Job 1 is under process while Job 2 is idle. Similarly moving vertically ~~moving~~ means Job 2 is under process while Job 2 is idle. The diagonal movement along this line shows that both the jobs are under process simultaneously.
- An optimal path is one that minimizes the idle time for both the ~~machines~~. jobs. Thus, we must choose a path in which the diagonal movement is maximum.
- Diagonal movement through rectangle areas is not possible.

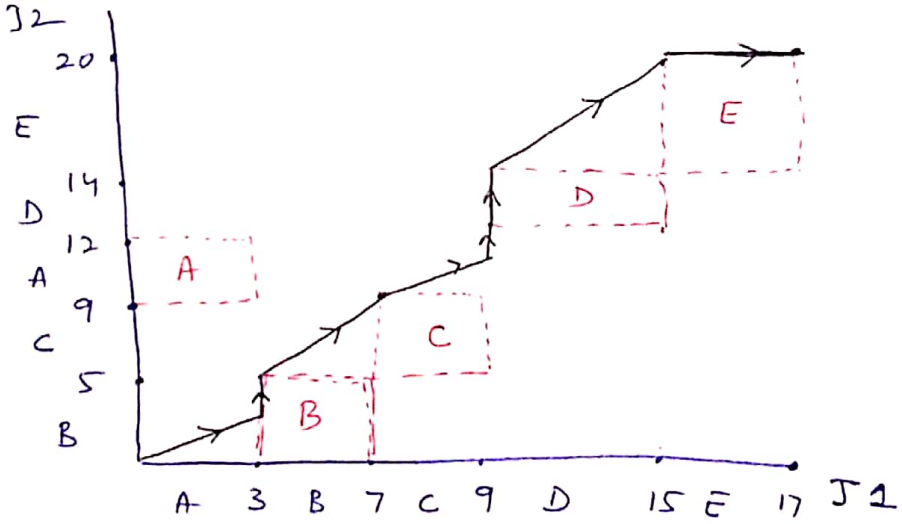
Imp.

Step 1 :

from the problem →



Step 2 : Draw blocks.



Step 3 : Draw the path. such that diagonal movement (with 45° with x-axis & y-axis) must be maximum. That will minimize the ~~total~~ elapsed time. [See the black marked line].

Elapsed time for Job 1 = 17 + Idle time for Job 1 → many Vertically

= 17 + 2 ('B') + 2+1 ('D')

= 22 Hrs.

Elapsed time for Job 2 = 20 + Idle time for Job 2 → many horizontally

= 20 + 2 → E

= 22 Hrs.