**e.g.**

Set of jobs for scheduling, using Greedy algorithm.

A

| Slots | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Job | Alpha | Beta | Gamma | Delta | Theta |
| Profit | 60 | 100 | 20 | 40 | 20 |
| Deadline | 2 | 1 | 3 | 2 | 1 |


B

| Slots | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Job | Beta | Alpha | Delta | Gamma | Theta |
| Profit | 100 | 60 | 40 | 20 | 20 |
| Deadline | 1 | 2 | 2 | 3 | 1 |

Create a vector (size = max. deadline, 3 in this case) of two elements profit and name of jobs. Initialize profit with 0 and name with XYZ.

C

| Index | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Job | XYZ | XYZ | XYZ | XYZ |
| Profit | 0 | 0 | 0 | 0 |

Table B slot 0 selected, and added in index 0 (deadline - 1) table C.

C

| Index | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Job | Beta | XYZ | XYZ | XYZ |
| Profit | 100 | 0 | 0 | 0 |

Table B slot 1 selected, and added in index 1 (deadline - 1) table C.

C

| Index | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Job | Beta | Alpha | XYZ | XYZ |
| Profit | 100 | 60 | 0 | 0 |

Table B slot 2 selected, and rejected because all the indices <= (deadline - 1) are already occupied by higher profit jobs.

Table B slot 3 selected, and added in index 2 (deadline - 1) table C.

C

| Index | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Job | Beta | Alpha | Gamma | XYZ |
| Profit | 100 | 60 | 20 | 0 |

Table B slot 4 selected, and rejected for the same reason.

Final set of jobs in the sequential order

C

| Index | 0 | 1 | 2 |
|---|---|---|---|
| Job | Beta | Alpha | Gamma |
| Profit | 100 | 60 | 20 |

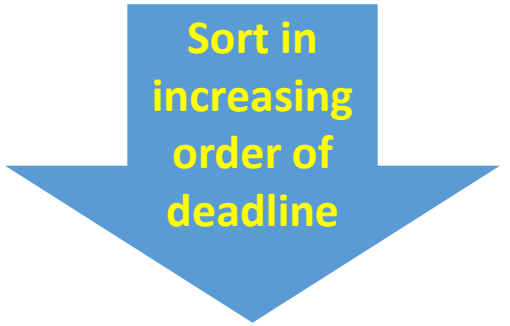
Maximum profit that can be earned
= 100 + 60 + 20 = 180

**e.g.**

Set of jobs for scheduling, using Brute Force algorithm.

X

| Slots | 0 | 1 | 2 |
|---|---|---|---|
| Job | Pi | Chi | Psi |
| Profit | 60 | 100 | 40 |
| Deadline | 2 | 1 | 2 |




Y

| Slots | 0 | 1 | 2 |
|---|---|---|---|
| Job | Chi | Pi | Psi |
| Profit | 100 | 60 | 40 |
| Deadline | 1 | 2 | 2 |

Generate all the possible solution sets.

| | A | B | C | | D |
|---|---|---|---|---|---|
| Index | 0 | 0 | 0 | 1 | 0 |
| Job | Chi | Pi | Chi | Pi | Psi |
| Profit | 100 | 60 | 100 | 60 | 40 |
| Deadline | 1 | 2 | 1 | 2 | 2 |

| E | | F | | G | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 2 |
| Chi | Psi | Pi | Psi | Chi | Pi | Psi |
| 100 | 40 | 60 | 40 | 100 | 60 | 40 |
| 1 | 2 | 2 | 2 | 1 | 2 | 2 |

Set A selected, and is feasible (index assigned <= deadline - 1 for each element of the set).
Current profit = 100 & current solution set = A

Set B selected, and is feasible. But rejected, because profit(B) < current profit.

Set C selected, and is feasible. Profit(C) > current profit.
Current profit and solution set updated.
Current profit = 160 & current solution set = C

Set D selected, and is feasible. But rejected, because profit (D) < current profit.

Same as set D, set E & F are rejected.

Set G selected, and rejected, because it is not feasible.

Final set of jobs in the sequential order

C

| Index | 0 | 1 |
|---|---|---|
| Job | Chi | Pi |
| Profit | 100 | 60 |

Maximum profit that can be earned
= 100 + 60 = 160