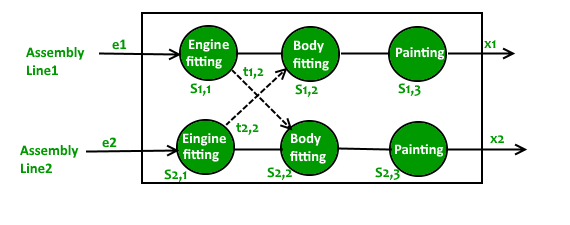
**Dynamic Programming | Set 34 (Assembly Line Scheduling)**

A car factory has two assembly lines, each with n stations. A station is denoted by Si,j where i is either 1 or 2 and indicates the assembly line the station is on, and j indicates the number of the station. The time taken per station is denoted by ai,j. Each station is dedicated to some sort of work like engine fitting, body fitting, painting and so on. So, a car chassis must pass through each of the n stations in order before exiting the factory. The parallel stations of the two assembly lines perform the same task. After it passes through station Si,j, it will continue to station Si,j+1 unless it decides to transfer to the other line. Continuing on the same line incurs no extra cost, but transferring from line i at station j – 1 to station j on the other line takes time ti,j. Each assembly line takes an entry time ei and exit time xi which may be different for the two lines. Give an algorithm for computing the minimum time it will take to build a car chassis.

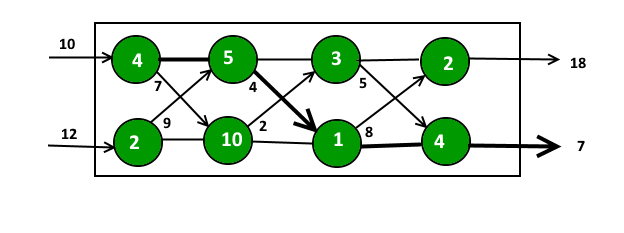
The below figure presents the problem in a clear picture:



The following information can be extracted from the problem statement to make it simpler:

* Two assembly lines, 1 and 2, each with stations from 1 to n.
* A car chassis must pass through all stations from 1 to n in order(in any of the two assembly lines). i.e. it cannot jump from station i to station j if they are not at one move distance.
* The car chassis can move one station forward in the same line, or one station diagonally in the other line. It incurs an extra cost ti, j to move to station j from line i. No cost is incurred for movement in same line.
* The time taken in station j on line i is ai, j.
* Si, j represents a station j on line i.

**Solution instance**:



**Minimum crossing time**:

35