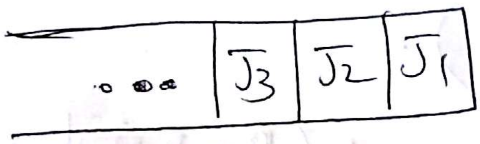
Shortest Job first (SJF)

$J = \{J_1, J_2, \ldots, J_n\}$

$\tau = \{t_1, t_2, \ldots, t_n\}$ t_i : task length



Schedule sequence : $\{J_1, J_2, \ldots, J_n\} \rightarrow W_s$
(arbitrary)

$\overline{W}_s$: Average waiting time of tasks

$$\overline{W}_s = (W_1 + W_2 + \cdots + W_n) \cdot \frac{1}{n} \quad \text{—— (1)}$$

Where, $W_k = W_{k-1} + t_{k-1}$ —— (2)

W_k: waiting time of task k

W_{k-1}: waiting time of task (k-1)

t_{k-1}: task length for $(k-1)^{st}$ task

Using (2) in (1) recursively,

$$\overline{W}_s = \left((n-1)t_1 + (n-2)t_2 + \cdots + t_n\right) \cdot \frac{1}{n}$$

$$\overline{W}_s = \frac{1}{n}\left((n-1)t_1 + (n-2)t_2 + \cdots + (n-k)t_k + \cdots + t_1\right) \longrightarrow (3)$$

Scanned by CamScanner

Consider two tasks $(k-j)$, k, $k>j$, in W_s such that $t_{k-j} > t_k$

We generate another schedule such that we swap tasks (in W_s) $k-j$ & k (such that task k is executed first).

$$W_s' = \left((n-1)t_1 + (n-2)t_2 + \cdots + (n-k+j)t_k + \cdots + (n-k)t_{k-j} + \cdots + t_n \right)\left(\tfrac{1}{n}\right) \quad (4)$$

$(3)-(4) \Rightarrow (W_s - W_s') =$

$$(n-k+j)t_{k-j} + (n-k)t_k - (n-k+j)t_k - (n-k)t_{k-j}$$

$$= j(t_{k-j} - t_k) > 0 \Rightarrow \underline{W_s' < W_s}$$ ■

Scanned by CamScanner