1 Deadlines are Marching

Example output from testing script where Int1 = 3, Int2 = 7. Any of the listed results were allowed, based on assumptions submissions made.

```
>>> running FIFO...
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 13.9 . TCT = 29
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 16.4 . TCT = 36
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 16.7 . TCT = 36
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 19.3 . TCT = 36
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 16.4 . TCT = 36
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 16.7 . TCT = 36
[ 0 4 7 5 8 1 2 6 9 3 ] . ACT = 19.3 . TCT = 36

>>> running LIFO...
[ 0 5 2 6 3 9 1 8 4 7 ] . ACT = 11.0 . TCT = 29
[ 0 5 2 3 6 9 1 8 4 7 ] . ACT = 15.8 . TCT = 36
[ 0 5 2 6 3 9 1 8 4 7 ] . ACT = 13.7 . TCT = 36
[ 0 5 2 6 3 9 1 8 4 7 ] . ACT = 12.7 . TCT = 36
[ 0 5 2 3 6 9 1 8 4 7 ] . ACT = 15.8 . TCT = 36
[ 0 5 2 6 3 9 1 8 4 7 ] . ACT = 13.7 . TCT = 36
[ 0 5 2 6 3 9 1 8 4 7 ] . ACT = 12.7 . TCT = 36

>>> running SJF...
[ 4 0 1 2 5 3 8 9 6 7 ] . ACT = 8.9 . TCT = 29
[ 4 0 1 5 3 8 9 2 6 7 ] . ACT = 11.4 . TCT = 36
[ 4 0 1 2 5 8 9 6 3 7 ] . ACT = 10.4 . TCT = 36
[ 0 5 2 1 3 9 6 4 8 7 ] . ACT = 12.0 . TCT = 36
[ 4 0 1 5 3 8 9 2 6 7 ] . ACT = 11.4 . TCT = 36
[ 4 0 1 2 5 8 9 6 3 7 ] . ACT = 10.4 . TCT = 36
[ 0 5 2 1 3 9 6 4 8 7 ] . ACT = 12.0 . TCT = 36

>>> running SRTF...
[ 4 0 1 2 3 5 8 9 6 7 ] . ACT = 8.9 . TCT = 29
[ 4 0 1 5 3 8 9 2 6 7 ] . ACT = 11.4 . TCT = 36
[ 4 0 1 2 5 8 9 6 3 7 ] . ACT = 10.4 . TCT = 36
[ 0 1 2 5 3 9 6 4 8 7 ] . ACT = 12.0 . TCT = 36
[ 4 0 1 5 3 8 9 2 6 7 ] . ACT = 11.4 . TCT = 36
[ 4 0 1 2 5 8 9 6 3 7 ] . ACT = 10.4 . TCT = 36
[ 0 1 2 5 3 9 6 4 8 7 ] . ACT = 12.0 . TCT = 36

>>> running EDF...
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 11.1 . TCT = 29
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 14.4 . TCT = 36
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 13.1 . TCT = 36
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 17.5 . TCT = 36
```

```
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 14.4 . TCT = 36
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 13.1 . TCT = 36
[ 4 0 8 1 2 5 7 6 3 9 ] . ACT = 17.5 . TCT = 36
```

2 Not a Mathematician

2.1 Proof of claim

Consider task set $\{t_1, t_2, ..., t_n\}$, l_i denotes the time it takes to finish t_i. Without lost of generality, we can assume $l_i \leq l_j$ for $i < j$.

Now assume there is an execution order that achieves better average completion time than SJF. Denote it as $t_{i_1}, t_{i_2}, ..., t_{i_n}$. Then in this order, there exists some $j < k$ such that $li_j > li_k$.

The average completion time of this execution will be

$$T_1 = \frac{1}{n}(nl_{i_1} + (n-1)l_{i_2} + ... + (n+1-j)l_{i_j} + ... + (n+1-k)l_{i_k} + ... + 2l_{i_{n-1}} + l_{i_n})$$

If we switch execution of t_{i_j} and t_{i_k} in this order, the average completion time will be

$$T_2 = \frac{1}{n}(nl_{i_1} + (n-1)l_{i_2} + ... + (n+1-j)l_{i_k} + ... + (n+1-k)l_{i_j} + ... + 2l_{i_{n-1}} + l_{i_n})$$

$$\begin{aligned} T_2 - T_1 &= \frac{1}{n}((k-j)l_{i_k} + (j-k)l_{i_j}) \\ &= \frac{1}{n}(j-k)(l_{i_j} - l_{i_k}) \end{aligned}$$

Because $j < k$, $l_{i_j} > l_{i_k}$, we have $T_2 - T_1 < 0$. So for any pair of tasks which executes in reverse order of requiring time, switching them will always reduce average completion time. So SJF leads to optimal average completion time.

2.2 Average Completion Time

Because we use SJF scheduling policy, t_i will be served before t_j for $i < j$. So these n tasks will be served in order from t_1 to t_n.

We can compute the completion time of t_i:

$$c_i = \sum_{j=1}^{i} l_i$$

Now we compute average completion time of T:

$$\begin{aligned} \bar{c} &= \sum_{i=1}^{n} c_i/n = \sum_{i=1}^{n}\sum_{j=1}^{i} l_i/n \\ &= \frac{nl_1 + (n-1)l_2 + ... + 2l_{n-1} + l_n}{n} \end{aligned}$$

2.3 Expectation of average of Monkey Scheduling

Example output from testing script where Int1 = 3, Int2 = 7.

```
remainSum = 61.0
ACT: 41.25
```

2.4 Expectation of average $n \geq 10$

Denote $s := \sum_{i=1}^{n} l_i$. Then

$$\begin{aligned}
E[\bar{c}] &= E[\sum_{i=1}^{n} c_i/n] = (\sum_{i=1}^{n} E[c_i])/n \\
&= (nl_{i_1} + (n-1)l_{i_2} + (n-2) \cdot \frac{s - l_{i_1} - l_{i_2}}{n-2} + ... + 2 \cdot \frac{s - l_{i_1} - l_{i_2}}{n-2} + \frac{s - l_{i_1} - l_{i_2}}{n-2})/n \\
&= \frac{1}{n}(nl_{i_1} + (n-1)l_{i_2} + \frac{\sum_{i=1, i\neq i_1, i\neq i_2}^{n} l_i}{n-2} \cdot \frac{(n-1)(n-2)}{2}) \\
&= \frac{1}{n}(nl_{i_1} + (n-1)l_{i_2} + \sum_{i=1, i\neq i_1, i\neq i_2}^{n} l_i \cdot \frac{n-1}{2})
\end{aligned}$$

3 Network 101

Denote `Int1` with i_1, and `Int2` with i_2.

3.1 One packet A $\rightarrow$ B

$$\frac{2}{3} \cdot 10^{-3} + \frac{2}{3} \cdot 10^{-2} + 10^{-2} + \frac{2}{3} \cdot 10^{-3} = (\frac{4}{3} \cdot 10^{-3} + \frac{5}{3} \cdot 10^{-2})s$$

At switch A:

$$\frac{10^4 + i_i \cdot 10^3}{10^6} = 10^{-2} + i_1 \cdot 10^{-3} = (1 + \frac{i_1}{10}) \cdot 10^{-2} s$$

At switch B:

$$\frac{10^4 + i_i \cdot 10^3}{500 \times 1000} = \frac{10 + i_1}{5} \cdot 10^{-2} = (2 + \frac{i_1}{5}) \cdot 10^{-2} s$$

At switch C, same as switch A. At switch D,

$$\frac{10^4 + i_i \cdot 10^3}{2 \cdot 10^6} = (0.5 + \frac{i_1}{20}) \cdot 10^{-2} s$$

Total time:

Store and forward

$$\frac{4}{3} \cdot 10^{-3} + \frac{5}{3} \cdot 10^{-2} + (1 + \frac{i_1}{10} + 2 + \frac{i_1}{5} + 1 + \frac{i_1}{10} + 0.5 + \frac{i_1}{20}) \cdot 10^{-2} = (6.3 + 0.45 i_1) \cdot 10^{-2} s$$

Forward immediately

$$\frac{4}{3} \cdot 10^{-3} + \frac{5}{3} \cdot 10^{-2} + (2 + \frac{i_1}{5}) \cdot 10^{-2} = (3.8 + 0.2i_1) \cdot 10^{-2}s$$

3.2 Sending one file

- How long # chunks: $500 + 50i_i$
  actual chunk size: 2040 bytes
  Total time:
  $$\frac{2}{3} \cdot 10^{-3} + 2040 \times 8 \times (500 + 50i_1)/10^6$$
- Goodput
  $$10^6 \cdot \frac{2000}{2040} = \frac{50}{51} \cdot 10^6$$

3.3 N P-bit packet

Packets dropping only happens at B.

3.3.1 If Store and Forward

processing time cannot be ignored:

- # packets dropped: $\lceil (N - 10)/2 \rceil$
- Index of dropped packets: $2i + 1$ for $i \geq 5$

or ignore processing time (the first bit of packet 6 leaves at exactly the same time the first bit of packet 11 arrives)

- # packets dropped: $\lfloor (N - 10)/2 \rfloor$
- Index of dropped packets: $2i$ for $i \geq 6$

3.3.2 If Forward Immediately

- # packets dropped: $\lfloor (N - 10)/2 \rfloor$
- Index of dropped packets: $2i$ for $i \geq 6$

Explanation: Bandwidth of B $\rightarrow$ C is half of A $\rightarrow$ B. So in the process one packet being sent from B, two packets arrives at B from A.

Assuming B begins sending a packet after receiving the whole packet. So at the time the whole packet 1 leaves B, packet 2 and 3 are in buffer;

at the time the whole packet 2 leaves B, packet 3, 4, 5 are in buffer;

3 leaves, 4, 5, 6, 7 in buffer;

4 leaves, 5, 6, 7, 8, 9 in buffer;

5 leaves, 6, 7, 8, 9, 10 in buffer. 11 wants to get in too, but buffer is full.

6 leaves, 7, 8, 9, 10, 12 in buffer. 13 dropped.

....

If B begining sends a packet as it receives the first bit, then

at the time the whole packet 1 leaves B, packet 2 is in buffer;
at the time the whole packet 2 leaves B, packet 3, 4 are in buffer;
3 leaves, 4, 5, 6 in buffer;
4 leaves, 5, 6, 7, 8 in buffer;
5 leaves, 6, 7, 8, 9, 10 in buffer;
6 leaves, 7, 8, 9, 10, 11 in buffer, 12 dropped;
7 leaves, 8, 9, 10, 11, 13 in buffer, 14 dropped;
....

3.4 Lost of maximum rate

3/4

4 The Furthest Distance in the World

Denote `Int1` with i_1, and `Int2` with i_2.

4.1 Time for one message

\# packet: $i_2 \mod 4 + 1$
Time for one packet:

$$\begin{aligned} T &= L \cdot (M/B) + 3L \times 10^{-3} \\ &= (3 + i_1 \mod 4) \cdot \frac{2000 + i_2 \cdot 100}{1000 + i_1 \cdot 100} + 3(3 + i_1 \mod 4) \times 10^{-3} \end{aligned}$$

Time for whole message:

$$\begin{aligned} T_m &= L \cdot (M/B) + 3L \times 10^{-3} \\ &= (i_2 \mod 4 + 1)((3 + i_1 \mod 4) \cdot \frac{2000 + i_2 \cdot 100}{1000 + i_1 \cdot 100} + 3(3 + i_1 \mod 4) \times 10^{-3}) \end{aligned}$$

4.2 Optimized time

Time for one packet:

$$\begin{aligned} T &= M/B + (L-1) \cdot (H/B) + 3L \times 10^{-3} \\ &= \frac{2000 + i_2 \cdot 100}{1000 + i_1 \cdot 100} + (2 + i_1 \mod 4)(\frac{100 + i_1 \cdot 10}{1000 + i_1 \cdot 100}) + 3(3 + i_1 \mod 4) \times 10^{-3} \end{aligned}$$

Time for whole message:

$$\begin{aligned} T_m &= (\#packets)(M/B + (L-1) \cdot (H/B)) + 3L \times 10^{-3} \\ &= (i_2 \mod 4 + 1)(\frac{2000 + i_2 \cdot 100}{1000 + i_1 \cdot 100} + (2 + i_1 \mod 4)(\frac{100 + i_1 \cdot 10}{1000 + i_1 \cdot 100}) + 3(3 + i_1 \mod 4) \times 10^{-3}) \end{aligned}$$

4.3 Virtual circuit

$$
\begin{aligned}
T_m &= L \cdot C/B + S/B + 3L \times 10^{-3} \\
&= ((3 + i_1 \mod 4) \cdot 0.8(2000 + i_2 \cdot 100) + (1900 + 100 i_2 - 10 i_1) \cdot (i_2 \mod 4 + 1))
\end{aligned}
$$