Problem 1 [30 points]: Consider a ring network with nodes 1, 2, ..., K. In this network, a customer that completes service at node \( i \) exits the network with probability \( p \), or it is routed to node \( i+1 \) with probability \( 1-p \), for \( i = 1, 2, ..., K-1 \). Customers that complete service at node \( K \), either exit the network, or are routed to node 1, with respective probabilities \( p \) and \( 1-p \).

At each node, external customers arrive according to a Poisson process with rate \( \gamma \). The service times at each node are exponentially distributed with rate \( \mu \). The arrival processes and the service times at the various nodes are independent.

1. [7 points] Find the aggregate arrival rates \( \lambda_i, i = 1, 2, ..., K \).
2. [2 points] Under what conditions does the ring network have a stationary distribution?
3. [7 points] Assuming that the conditions of question 2 are satisfied, find the stationary distribution of the network.
4. [7 points] Find the average time that a customer spends in the network.
5. [7 points] Is this ring network reversible? Justify your answer.

Solution 1: The network is shown in the following figure. Since there are external arrivals, it is an open Jackson network.

1. The aggregate arrival rates satisfy:
1. Since all nodes are symmetric – same arrival rates, service rates, and routing probabilities – the aggregate arrival rates must be equal: \( \lambda_1 = \lambda_2 = \ldots = \lambda_K = \lambda \). The above equations, then, give:

\[
\lambda = \gamma + (1 - p)\lambda_k
\]

2. The network has a stationary distribution if the aggregate arrival rate at each node is less than the service rate at the node. Therefore, we must have:

\[
\frac{\gamma}{p} < \mu
\]

3. From Jackson’s theorem for open networks, the stationary distribution is:

\[
p(n_1, \ldots, n_K) = \prod_{i=1}^{K} p_i(n_i) = \prod_{i=1}^{K} (1 - \rho)\rho^n_i = (1 - \rho)^K \rho^{n_i + \cdots + n_K}, \quad \rho = \frac{\gamma}{p\mu}
\]

4. The average number of customers at each queue is:

\[
N_i = \frac{\rho}{1 - \rho}, \quad i = 1, \ldots, K
\]

and the average number of customers in the network:

\[
N = K \frac{\rho}{1 - \rho}
\]

Since the arrival rate at the network is \( K\gamma \), Little’s theorem implies that the average time a customer spends in the network is:

\[
T = \frac{N}{K\gamma} = \frac{1}{\gamma} \frac{1}{1 - \rho} = \frac{1}{p\mu - \gamma}
\]

5. The network is not time reversible. To see this, note that a transition of a customer from node \( i \) to node \( i+1 \) in the original network corresponds to a transition from node \( i+1 \) to node \( i \) in the reversed network. Considering state \( n \) with \( n_i > 0 \), in the original network, we have:

\[
q(n, n - e_i + e_{i+1}) = \mu(1 - p) > 0
\]

while in the reversed network:

\[
q^*(n, n - e_i + e_{i+1}) = 0
\]

since customers cannot move from node \( i \) to node \( i+1 \) in the reversed network.
Problem 2 [20 points]: Consider a closed Jackson network with $K$ nodes and $M$ customers. The normalization constant for the network is:

$$G(M) = \sum_{n_1 + \ldots + n_K = M} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_K^{n_K}$$  \hspace{1cm} (1)$$

1. [3 points] Show that the number of terms in the summation on the right-hand side of eq. (1) is

$$\binom{M + K - 1}{M}$$

2. [7 points] Show that the normalization constant can be calculated based on the following iterative algorithm (Buzen’s algorithm):

$$G(m,k) = G(m,k-1) + \rho_k G(m-1,k), \quad m = 0,1,\ldots,M, \ k = 1,\ldots,K \hspace{1cm} (2)$$

with initial conditions:

$$G(m,1) = \rho_1^m, \quad m = 0,\ldots,M$$

$$G(0,k) = 1, \quad k = 1,\ldots,K$$

3. [5 points] Let $x_i$ be the number of customers at node $i$ at steady state. Prove that:

$$P\{x_i \geq m\} = \rho_i^m \frac{G(M-m)}{G(M)}, \quad m = 1,2,\ldots,M$$

4. [5 points] Prove that the average throughput of node $i$ is:

$$\gamma_i(M) = \lambda_i \frac{G(M-1)}{G(M)} \hspace{1cm} (3)$$

Solution 2:

1. The number of terms in the summation is equal to the number of possible states $(n_1,\ldots,n_K)$, such that $n_1 + \cdots + n_K = M$. Note that each state corresponds to a way of distributing the M customers (“balls”) in the K queues (“boxes”). Recalling that there are

$$\binom{M + K - 1}{M} = \binom{M + K - 1}{K - 1}$$

ways to distribute M balls in K boxes, the result follows.

2. For $m > 0$ and $k > 1$ we split the sum into two sums over disjoint sets of indices, corresponding to $n_k = 0$, and $n_k > 0$.

$$G(m,k) = \sum_{n_1 + \ldots + n_k = m \atop n_k \geq 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k}$$

$$= \sum_{n_1 + \ldots + n_k = m \atop n_k = 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k} + \sum_{n_1 + \ldots + n_k = m \atop n_k > 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k}$$

$$= \sum_{n_1 + \ldots + n_k = m \atop n_k > 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_{k-1}^{n_{k-1}} + \sum_{n_1 + \ldots + n_k = m \atop n_k > 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k}$$
Note that the first sum is \( G(m, k-1) \). For the second sum, observing that \( n_k > 0 \), we define \( n_k = n_k' + 1 \), where \( n_k' \geq 0 \). Then:

\[
\sum_{n_1 + \cdots + n_k = m, \ n_k > 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k} = \sum_{n_1 + \cdots + n_k' + 1 = m, \ n_k' \geq 0} \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_k^{n_k'} = \rho_k G(m-1, k)
\]

Therefore: \( G(m, k) = G(m, k-1) + \rho_k G(m-1, k) \)

3. We have:

\[
P\{x_j \geq m\} = \sum_{n_1+\cdots+n_k=M, \ n_j \geq m} p(n) = \sum_{n_1+\cdots+n_k=M, \ n_j \geq m} \frac{\rho_1^{n_1} \cdots \rho_j^{n_j+m} \cdots \rho_k^{n_k}}{G(M)}
\]

\[
= \sum_{n_1+\cdots+n_k'+m = M, \ n_j' + n_j = m} \frac{\rho_1^{n_1'} \cdots \rho_j^{n_j+m} \cdots \rho_k^{n_k}}{G(M)}
\]

\[
= \frac{\rho_j^m}{G(M)} \sum_{n_j \geq 0} \rho_1^{n_1} \cdots \rho_j^{n_j} \cdots \rho_k^{n_k} \]

\[
= \frac{\rho_j^m}{G(M)} G(M-m)
\]

4. For the average throughput of node \( i \):

\[
\gamma_i(M) = \mu_i P\{x_i \geq 1\} = \mu_i \cdot \frac{G(M-1)}{G(M)} = \lambda_i \cdot \frac{G(M-1)}{G(M)}
\]
Problem 3 [20 points]: Consider the closed Jackson network of Figure 1, with $K=3$ nodes and $M$ customers.

![Figure 1](image1.png)

1. [5 points] Calculate the normalization constant $G(M)$ explicitly, using eq. (1).

2. [5 points] Find the average throughput $\gamma_i(M)$ of each node $i$ as a function of the number of customers $M$.

3. [5 points] Use Buzen’s algorithm, described by eq. (2), to calculate $G(M)$ and $\gamma_i(M)$, for $M = 1,2,3,4,5$.

4. [5 points] Consider the closed network of Figure 2, with $K=3$ nodes and $M$ customers.

![Figure 2](image2.png)

Show that this new network has the same stationary distribution with the network shown in Figure 1.

[Note: You can answer this question without calculating the stationary distributions explicitly.]

Solution 2: The “visit ratios” at the three queues are equal: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$. For a closed network these rates can only be determined to the extent of a multiplicative constant. Therefore, we can set $\lambda = \mu$. Then:

$$\rho_1 = \rho_2 = 1, \quad \rho_3 = \frac{1}{2}$$
1. The normalization constant is:

\[ G(M) = \sum_{n_1+n_2+n_3=M} \rho_1^{n_1} \rho_2^{n_2} \rho_3^{n_3} = \sum_{n_1+n_2+n_3=M} \frac{1}{2^{n_3}} = \sum_{n_1+n_2=M-n_3} \frac{1}{2^{n_3}} = \sum_{n_3=0}^{M} (M-n_3+1) \frac{1}{2^{n_3}} \]

(4)

Similarly to question 1 in Problem 2, the sum over \( n_1 + n_2 = M - n_3 \) in the above derivation has

\[ \binom{M - n_3 + (2 - 1)}{M - n_3} = (M - n_3 + 1) \]

terms. From eq. (4), we have:

\[ G(M) = (M + 1) \sum_{n_3=0}^{M} \frac{1}{2^{n_3}} - \frac{1}{2} \sum_{n_1+n_2=M-n_3}^{M} \frac{1}{2^{n_3}} \]

(5)

To compute the second series, note that for any \( x \), we have:

\[ \sum_{n=1}^{N} nx^{n-1} = \frac{\partial}{\partial x} \sum_{n=0}^{N} x^n = \frac{\partial}{\partial x} \left( \frac{1-x^{N+1}}{1-x} \right) = \frac{N x^{N+1} - (N+1)x^N + 1}{(x-1)^2} \]

Then, eq. (5) gives:

\[ G(M) = (M + 1) \frac{1-(1/2)^{M+1}}{1-1/2} - \frac{1}{2} \frac{M(1/2)^{M+1} - (M + 1)(1/2)^M + 1}{(1-1/2)^2} \]

\[ = \frac{1+2^{M+1}M}{2^M} = 2^{-M} + 2M \]

2. Using eq. (3), the average throughput of node \( i \) is:

\[ \gamma_i(M) = \lambda_i \frac{G(M-1)}{G(M)} = \mu \frac{2^{-M+1} + 2(M-1)}{2^{-M} + 2M}, \quad i = 1, 2, 3 \]

3. Using Buzen's algorithm, we can find the value of \( G(M) \) for a given \( M \), iteratively, as described by eq. (2). The average throughput of the various queues can be found from \( G(M-1) \) and \( G(M) \), using eq. (3). The iterative calculation is organized in the following tables. The first table shows the values \( G(m,k) \) calculated by Buzen's algorithm, while the second gives the normalization constant and the throughput for \( M=0,1,2,3,4,5 \).
<table>
<thead>
<tr>
<th>$m$</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>M</th>
<th>$G(M)$</th>
<th>$\gamma(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$\frac{5}{2}$</td>
<td>1</td>
<td>$\frac{5}{2}$</td>
<td>$\frac{2}{5^\mu}$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>$\frac{17}{4}$</td>
<td>2</td>
<td>$\frac{17}{4}$</td>
<td>$\frac{10}{17^\mu}$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>$\frac{49}{8}$</td>
<td>3</td>
<td>$\frac{49}{8}$</td>
<td>$\frac{34}{49^\mu}$</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>$\frac{129}{16}$</td>
<td>4</td>
<td>$\frac{129}{16}$</td>
<td>$\frac{98}{129^\mu}$</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>$\frac{321}{32}$</td>
<td>5</td>
<td>$\frac{321}{32}$</td>
<td>$\frac{86}{107^\mu}$</td>
</tr>
</tbody>
</table>

4. For the closed network of Figure 2, the flow conservation equations are

$$\lambda_1 = \frac{\lambda_3}{2}, \quad \lambda_2 = \frac{\lambda_3}{2}, \quad \lambda_3 = \lambda_1 + \lambda_2$$

Taking $\lambda_1 = \mu$, we have:

$$\rho_1 = \rho_2 = 1, \quad \rho_3 = \frac{1}{2}$$

which are equal to the $\rho_i$'s of the nodes in the network shown in Figure 1. Therefore, the two networks have the same stationary distribution:

$$p(n_1, n_2, n_3) = \frac{\rho_{1}^{n_1} \rho_{2}^{n_2} \rho_{3}^{n_3}}{\sum_{n_1 + n_2 + n_3 = M} \rho_{1}^{n_1} \rho_{2}^{n_2} \rho_{3}^{n_3}}$$
Problem 4 [30 points]: Consider the network of Figure 3, which is represented by a directional graph, with cost (length) $d_{ij}$ associated with each directed link $(i, j)$.

![Figure 3](image-url)

1. [20 points] Find the shortest distance from every node in the network to destination node A, using two different shortest path algorithms – Bellman-Ford, Dijkstra, or Floyd-Warshall. For each algorithm, indicate clearly the operations performed at each iteration of the algorithm.

2. [10 points] The network is transformed by replacing each pair of directed links with a single bidirectional link. The cost (weight) of the bidirectional link is equal to the maximum of the costs of the two original directional links it replaces. Find a minimum weight spanning tree, using Prim’s algorithm. Indicate clearly all steps in the construction of the minimum weight spanning tree.

Solution 4:

1. Since link (C,E) has negative length, we cannot use Dijkstra's algorithm for the computation of the shortest distances – the algorithm can only be used in networks with nonnegative lengths. Therefore, we will use (a) the Bellman-Ford, and (b) the Floyd-Warshall algorithms to calculate the shortest distances of all nodes to destination node A. In fact, Floyd-Warshall algorithm will determine the shortest distance between all pair of nodes $i$ and $j$ in the network.

**Bellman-Ford Algorithm:** The algorithm iterates on the maximum number of hops (links) $h$ that are allowed on a walk from any node to destination node A. At iteration $h$, the algorithm determines the shortest distance $D_i^h$ of node $i$ to node A, when walks of at most $h$ hops are considered – for all $i \neq 1$. The algorithm performs the following iteration:

$$D_i^{h+1} = \min_{j} \{d_{ij} + D_j^h\}, \quad i \neq 1$$

$$D_1^{h+1} = 0$$

with initial conditions:

$$D_i^0 = \infty, \quad i \neq 1$$

$$D_1^0 = 0$$
The Bellman-Ford algorithm determines the correct shortest distances after at most N-1 iterations if and only if, there are no cycles with negative length – a condition that is satisfied by the network under consideration.

The operations performed at each iteration of the algorithm can be organized in a tabular formulation, where the entries at row $h$ are $D_i^h$, $i = A, B, ..., G$, i.e., the shortest distances of all nodes to destination node A, over paths with at most $h$ hops. Since all cycles in the network have positive lengths, any shortest ($\leq h$) – walk cannot contain any cycles and is therefore a path.

The algorithm terminates after 6 iterations, since $D_i^6 = D_i^5$, $i = A, B, ..., G$. These are the shortest distances from the corresponding nodes to node A.

**Floyd-Warshall:** The Floyd-Warshall algorithm computes the shortest distances between every pair of nodes $i$ and $j$. The algorithm iterates on the set of nodes that can be used as intermediate nodes along the paths between two nodes, augmenting that set by one node at each iteration. Assuming that the nodes are labeled as 1, 2, ..., $N$ and that they are considered as intermediate nodes according these labels, then at iteration $n$, only nodes $\{1, ..., n\}$ can be used as intermediate nodes. At iteration $n+1$, node $n+1$ is also considered as intermediate node, and the algorithm performs the update:

$$D_{ij}^{n+1} = \min\{D_{ij}^n, D_{i,n+1}^n + D_{n+1,j}^n\}, \quad \forall i, j$$

where $D_{ij}^n$ is the shortest distance from $i$ to $j$ when only nodes 1, 2, ..., $n$ can be used as intermediate nodes. The evolution of the algorithm can be represented by a series of matrices. The $n^{th}$ matrix gives $D_{ij}^n$ for all nodes $i$ and $j$.

- $n = 0$; Intermediate nodes = {}
- $n = 1$; Intermediate nodes = \{A\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>7</td>
<td>$\infty$</td>
<td>3</td>
<td>4</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>$\infty$</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>$\infty$</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>-5</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>$\infty$</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

- $n = 2$; Intermediate nodes = \{A,B\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>$\infty$</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>$\infty$</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>-5</td>
<td>$\infty$</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

- $n = 3$; Intermediate nodes = \{A,B,C\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>$\infty$</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>$\infty$</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>-5</td>
<td>$\infty$</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>-3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>-1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

- $n = 4$; Intermediate nodes = \{A,B,C,D\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>-3</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>-5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>-3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>-1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

10 of 15
- $n = 5$; Intermediate nodes = \{A,B,C,D,E\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>-5</td>
<td>-4</td>
<td>-2</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- $n = 6$; Intermediate nodes = \{A,B,C,D,E,F\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>-5</td>
<td>-4</td>
<td>-2</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- $n = 7$; Intermediate nodes = \{A,B,C,D,E,F,G\}

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-5</td>
<td>-4</td>
<td>-2</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
2. Consider now the undirected graph with bidirectional links of Figure 4.

We use Prim’s Algorithm to construct a minimum weight spanning tree (MST). The algorithm starts with an arbitrary node as the initial fragment. At each iteration, the fragment is augmented by adding an adjacent link with minimum weight that does not create a cycle. Assume that the MST algorithm starts with node A as the initial fragment. The construction of the MST is illustrated in the following figures.

$n = 1$
- $n = 2$

![Diagram for $n = 2$](image1.png)

- $n = 3$

![Diagram for $n = 3$](image2.png)

- $n = 4$

![Diagram for $n = 4$](image3.png)

- $n = 5$

![Diagram for $n = 5$](image4.png)
$n = 6$
Bellman-Ford Algorithm:

\[ D_{i}^{k+1} = \min_{j} \{ d_{ij} + D_{j}^{k} \}, \quad i \neq 1 \]
\[ D_{1}^{k+1} = 0 \]

with initial conditions:

\[ D_{i}^{0} = \infty, \quad i \neq 1 \]
\[ D_{1}^{0} = 0 \]

Dijkstra’s Algorithm:

Initialization: \( P = \{1\}, \quad D_{1} = 0, \quad D_{j} = d_{j1}, \forall j \neq 1 \)

Iteration step 1: Find \( i \notin P \), such that \( D_{i} = \min_{j \in P} D_{j} \). Set \( P := P \cup \{i\} \).

Iteration step 2: For all \( j \notin P \), set: \( D_{j} := \min\{D_{j}, d_{ji} + D_{i}\} \).

Floyd-Warshall Algorithm:

\[ D_{ij}^{k+1} = \min\{D_{ij}^{k}, D_{ij}^{k-1} + D_{i}^{k}, D_{i}^{k+1}, j \}, \quad i \neq j \]

with initial conditions:

\[ D_{ij}^{0} = d_{ij}, \quad i \neq j \]

Average number of customers in an M/M/1 queue:

\[ N = \frac{\rho}{1 - \rho}, \quad \rho = \frac{\lambda}{\mu} \]

Other useful formulas:

\[ \sum_{n=0}^{N} x^{n} = \frac{1 - x^{N+1}}{1 - x} \]
\[ \sum_{n=1}^{N} nx^{n-1} = \frac{\partial}{\partial x} \sum_{n=0}^{N} x^{n} \]