No:
Name:

**GRAPHICS**

1. (15P) Use Dijkstra’s algorithm to find the length of a shortest path and a shortest path from a to z in the following weighted graph (show each iteration in below box).

2. (10P) For the following graph, determine if it has an Euler path that is not a circuit.

   It has an Euler trail that is not a circuit because it has two vertices with odd degree (b and e).

**TREES**

3. (15P) Construct an optimal Huffman code for the set of letters in the table. Find the average length of bit strings encoding 39-letter words with the Huffman code.

<table>
<thead>
<tr>
<th>letter</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>18</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
</tr>
</tbody>
</table>

   Average length = \( \frac{2 \times 4 + 12 \times 2 + 3 \times 4 + 18 \times 1 + 4 \times 3}{39} = 1.9 \)

4. In the following graph with its vertices in alphabetical order find a spanning tree using
   i. (10P) Breadth-first search.
   ii. (10P) Depth-first search.
AUTOMATA

1. According to finite state automaton transition diagram given on the right,
   i. (10P) Design the grammar rules.
   ii. (10P) Describe acceptable strings as a sentence.

\[
\begin{align*}
A & \rightarrow aB \mid bC & B & \rightarrow bF & C & \rightarrow aF \\
F & \rightarrow aB \mid bC \mid \lambda
\end{align*}
\]
Then by removing \(\lambda\),
\[
\begin{align*}
A & \rightarrow aB \mid bC & B & \rightarrow bF \mid b \\
C & \rightarrow aF \mid a & F & \rightarrow aB \mid bC
\end{align*}
\]

2. Let \(G\) be the grammar of a language with non terminal symbols \(\{E,F\}\), terminal symbol \(\{a,b,+,*\}\), starting symbol \(E\), and rules as follow.

\[
\begin{align*}
E & \rightarrow F & E & \rightarrow +FE \\
E & \rightarrow * FE & F & \rightarrow a \\
F & \rightarrow b
\end{align*}
\]

i. (10P) Find a derivation for the string " \(* +a * bba" .
ii. (10P) Draw its deterministic finite state automaton transition diagram by using non-deterministic one.

i.
It cannot be possible to derivate that string " \(* +a * bba" by using \(G\) grammar.

ii. (first non-deterministic one) (then deterministic one)