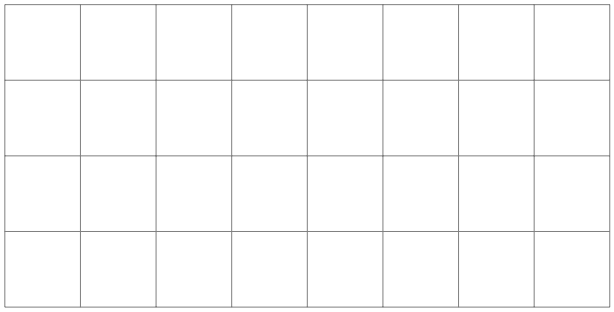
LaTeX: Undirected and directed graphs

Dr. Yihsiang Liow (September 17, 2020)

Contents

1 Grid

You can draw a helper grid to help you draw your picture. After you are done, you can remove the grid.

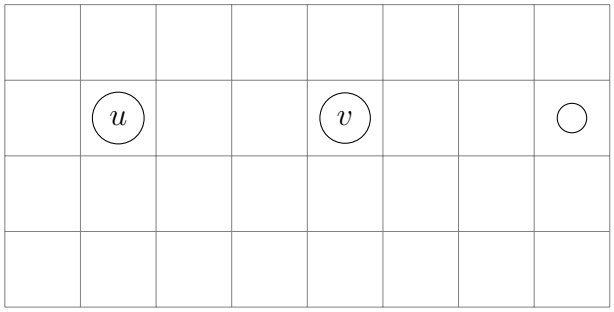


LaTeX code

```
You can draw a helper grid to help you draw your picture.
After you are done, you can remove the grid.
\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\end{tikzpicture}
\end{center}
```

2 Node

This is now you draw nodes. The `x`, `y`, `z` in the code are names of the nodes and will be used for drawing edges. Names are optional if you don't intend to refer to the nodes.



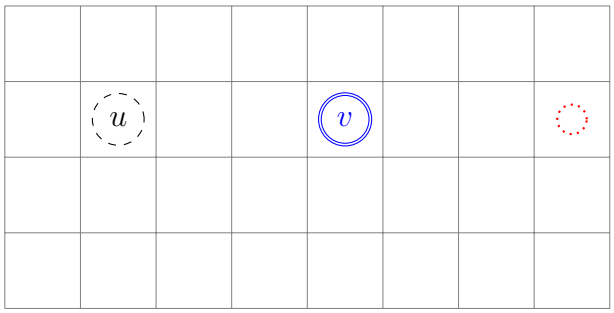
LaTeX code

```
This is now you draw nodes.
The \verb!x!, \verb!y!, \verb!z! in the code
are names of the nodes and will be used for drawing edges.
Names are optional if you don't intend to refer to the nodes.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\end{tikzpicture}
\end{center}
```

3 Node: boundary

You can change the boundary. You'll notice that for the double boundary case, the two lines are very close. I usually specify

`node[circle, draw, double, double distance=1pt]`

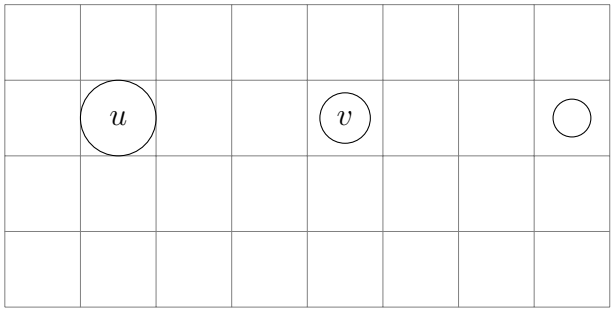


LaTeX code

```
You can change the boundary.
You'll notice that for the double boundary case, the two lines are very close.
I usually specify
\[
\verb!node[circle, draw, double, double distance=1pt]!
\]

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw[]           (-2.5,  0.5) node[circle, draw, dashed](x) {$u$};
\draw[thin, blue] ( 0.5,  0.5) node[circle, draw, double](y) {$v$};
\draw[thick, red] ( 3.5,  0.5) node[circle, draw, dotted](z) {};

\end{tikzpicture}
\end{center}
```

4 Node: size

You can control the size. The node will expand if necessary to contain the label. So you might need to adjust the size to make the sizes the same.

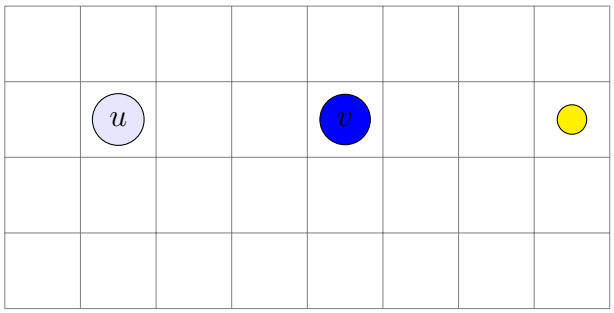


LaTeX code

```
You can control the size.
The node will expand if necessary to contain the label.
So you might need to adjust the size to make the sizes the same.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5,  0.5) node[circle, draw, minimum width=1cm](x) {$u$};
\draw ( 0.5,  0.5) node[circle, draw, minimum width=0.5cm](y) {$v$};
\draw ( 3.5,  0.5) node[circle, draw, minimum width=0.5cm](z) {};

\end{tikzpicture}
\end{center}
```

5 Node: color

You can change the bacground color:

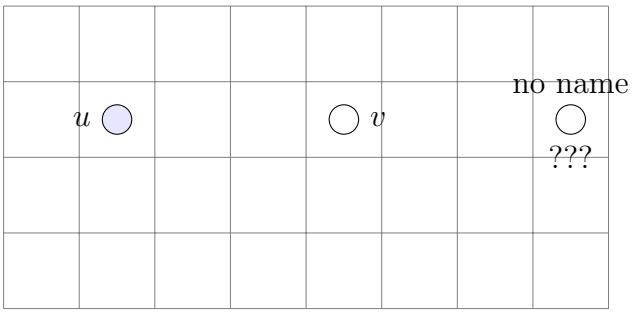


LaTeX code

```
You can change the bacground color:

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw, fill=blue!10](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw, fill=blue](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw, fill=yellow](z) {};

\end{tikzpicture}
\end{center}
```

6 Node: label

You can label the nodes outside. This is useful when your label is too huge for the node.

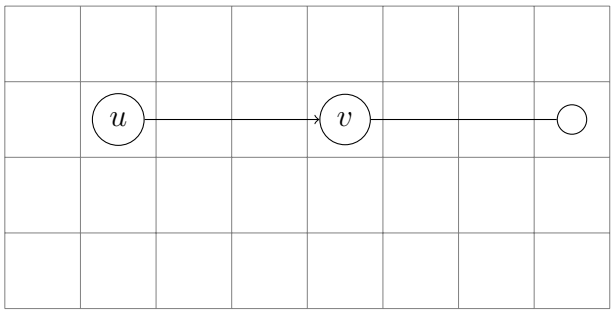


LaTeX code

```
You can label the nodes outside. This is useful when your label
is too huge for the node.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);


\draw (-2.5, 0.5) node[circle, draw, fill=blue!10,
                       label=left:{$u$}](x)                  {};
\draw ( 0.5, 0.5) node[circle, draw, label=right:{$v$}](y)   {};
\draw ( 3.5, 0.5) node[circle, draw,
                       label=above:{no name},
                       label=below:{???}](z)                 {};

\end{tikzpicture}
\end{center}
```

7 Edge

Now for edges ... you can have directed or undirected edges.

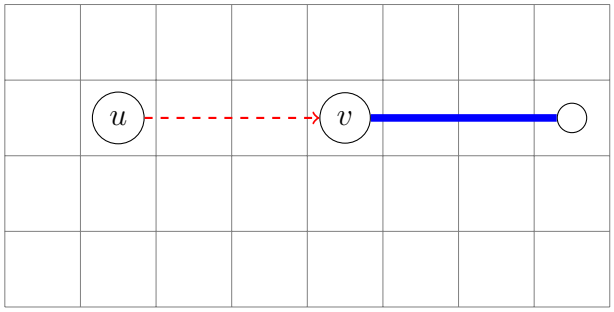


LaTeX code

```
Now for edges ... you can have directed or undirected edges.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\draw[->] (x) -- (y);
\draw     (y) -- (z);

\end{tikzpicture}
\end{center}
```

8 Edge

You can color the edges, change the thickness, and change the line style. Besides specifying the line width, there are also shorthands such as `thick` and `very thick`.

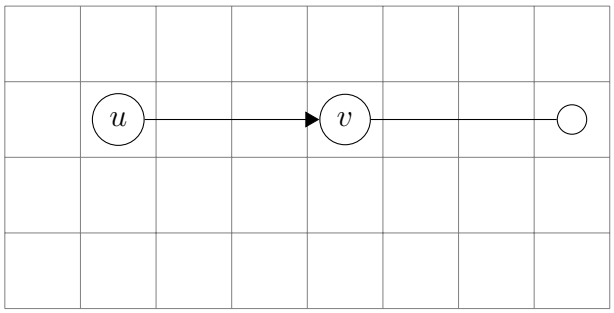


LaTeX code

```
You can color the edges, change the thickness, and change the line style.
Besides specifying the line width, there are also shorthands
such as \verb!thick! and \verb!very thick!.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\draw[->, dashed, red, thick] (x) -- (y);
\draw[blue, line width=0.1cm] (y) -- (z);

\end{tikzpicture}
\end{center}
```

9 Edge: arrow tip

You can change the arrow tip to a larger one. I find the default too small.

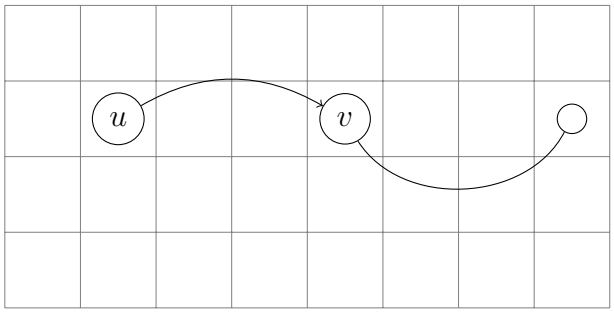


LaTeX code

```
You can change the arrow tip to a larger one. I find the default too small.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\draw[->, >=triangle 60] (x) -- (y);
\draw                    (y) -- (z);

\end{tikzpicture}
\end{center}
```

10 Edge: bending an edge

You can bend an edge. You replace the "`--`" with "`to[bend right=30]`" where 30 is the angle (in degrees).



LaTeX code

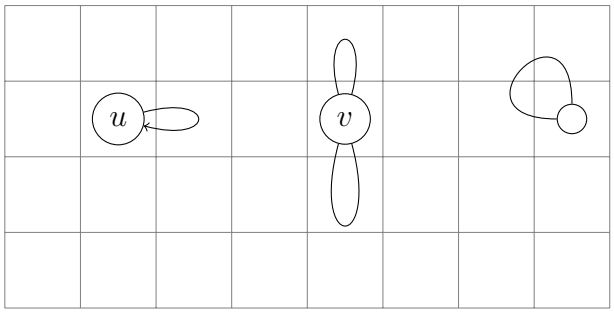
```
You can bend an edge.
You replace the \lq\lq\verb!--!" with \lq\lq\verb!to[bend right=30]!"
where 30 is the angle (in degrees).

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\draw[->] (x) to[bend left=30] (y);
\draw     (y) to[bend right=60] (z);

\end{tikzpicture}
\end{center}
```

11 Edge: loop

You can do loops.

You can specify the angles (in degrees) going out and coming in: `edge[in=30,out=60,loop]`. `distance` is of course how far the loop reaches.

There seems to be a bug in the pgf/tikz code. You must specify the `distance` in order for the arrow to disappear.



LaTeX code

```
You can do loops.

You can specify the angles (in degrees) going out and coming in:
\verb!edge[in=30,out=60,loop]!.
\verb!distance! is of course how far the loop reaches.

There seems to be a bug in the pgf/tikz code. You must specify the
\verb!distance! in order for the arrow to disappear.

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\draw [->] (x) edge[loop right, distance=1cm]       (x);
\draw      (y) edge[loop above, distance=1cm]       (y);
\draw      (y) edge[loop below, distance=1.5cm]     (y);
```

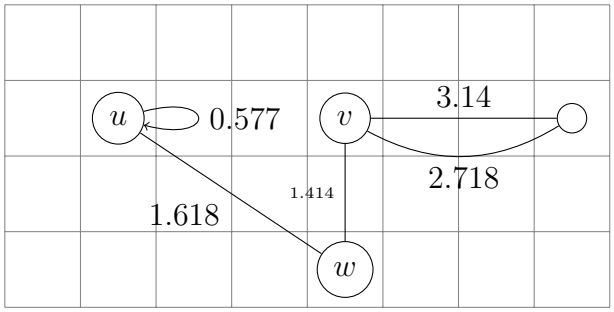
```
\draw      (z) edge[out=90, in=180, distance=1.5cm] (z);

\end{tikzpicture}
\end{center}
```

12 Edge: label

You can put labels on edges. The following show you how to label different edges. The basic idea is you create a `node` next to "`edge`", "`--`", or "`to`".

The following are the placement options for the label:

- `left`, `right`, `above`, `below`
- `above left`, `above right`, `below left`, `below right`



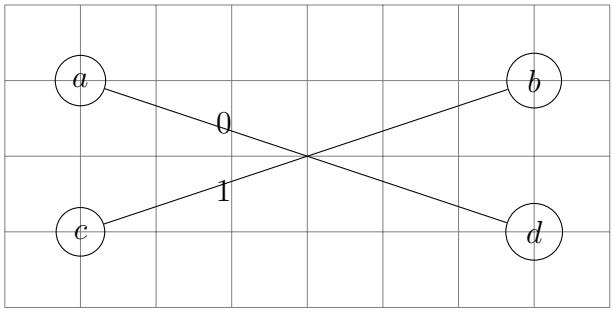
LaTeX code

```
You can put labels on edges.
The following show you how to label different edges.
The basic idea is you create a \verb!node! next to
\lq\lq \verb!edge!",
\lq\lq \verb!--!", or
\lq\lq \verb!to!".

The following are the placement options for the label:
\begin{itemize}
\li \verb!left!, \verb!right!, \verb!above!, \verb!below!
\li \verb!above left!, \verb!above right!, \verb!below left!, \verb!below right!
\end{itemize}

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
```

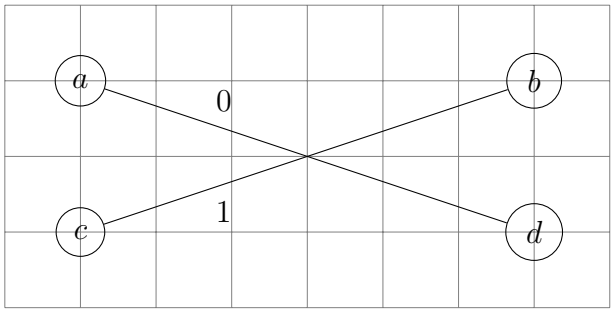
```
\draw ( 3.5, 0.5) node[circle, draw](z) {};
\draw ( 0.5,-1.5) node[circle, draw](w) {$w$};

\draw [->] (x) edge[loop right, distance=1cm] node[right]{0.577}        (x);
\draw      (y) --                             node[above]{3.14}         (z);
\draw      (y) to[bend right=30]              node[below]{2.718}        (z);
\draw      (y) --                             node[left]{{\tiny 1.414}} (w);
\draw      (x) --                             node[below left]{1.618}   (w);

\end{tikzpicture}
\end{center}
```

13 Edge: label position

Sometimes you need to move a label using `pos`:



If the label touches the edge you can do this:



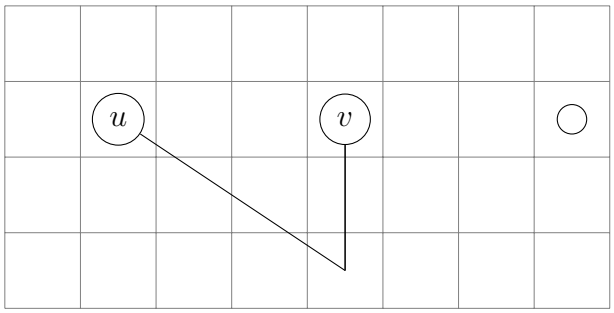
LaTeX code

```
Sometimes you need to move a label using \verb!pos!:
\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-3,  1) node[circle, draw](a) {$a$};
\draw ( 3,  1) node[circle, draw](b) {$b$};
\draw (-3, -1) node[circle, draw](c) {$c$};
\draw ( 3, -1) node[circle, draw](d) {$d$};

\draw (a) -- node[right, pos=0.25]{0} (d);
\draw (c) -- node[right, pos=0.25]{1} (b);

\end{tikzpicture}
\end{center}
```

```
If the label touches the edge you can do this:
\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-3,  1) node[circle, draw](a) {$a$};
\draw ( 3,  1) node[circle, draw](b) {$b$};
\draw (-3, -1) node[circle, draw](c) {$c$};
\draw ( 3, -1) node[circle, draw](d) {$d$};

\draw (a) -- node[right, pos=0.25, above right]{0} (d);
\draw (c) -- node[right, pos=0.25, below right]{1} (b);

\end{tikzpicture}
\end{center}
```

14 Coordinate

You can create a point (a coordinate).

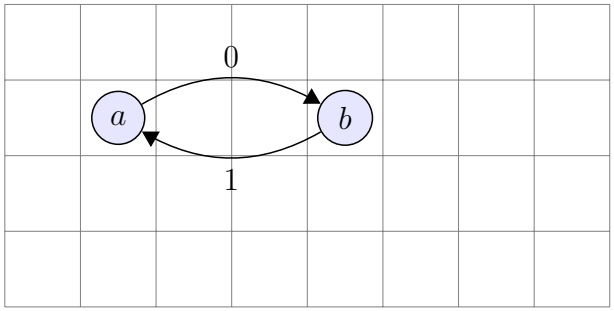


LaTeX code

```
You can create a point (a coordinate).

\begin{center}
\begin{tikzpicture}

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[circle, draw](x) {$u$};
\draw ( 0.5, 0.5) node[circle, draw](y) {$v$};
\draw ( 3.5, 0.5) node[circle, draw](z) {};

\coordinate (A) at (0.5, -1.5);

\draw (A) -- (x);
\draw (A) -- (y);
\draw (A) -- (y);

\end{tikzpicture}
\end{center}
```

15 Environment

You can specify defaults in the tikz environment (the block) and you can also define "styles" for reuse.

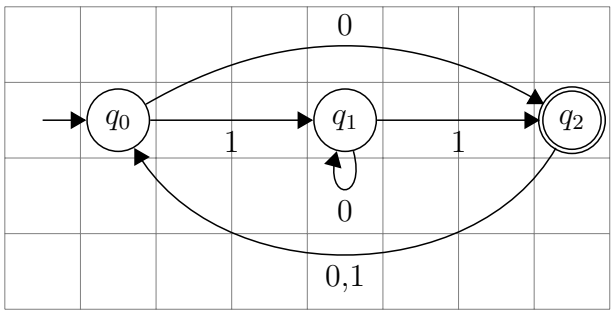


LaTeX code

```
You can specify defaults in the tikz environment (the block)
and you can also define \lq\lq styles" for reuse.

\begin{center}
\begin{tikzpicture}[
    >=triangle 60,       % default arrow tip
    line width=0.02cm,   % default line width
    bluecirc/.style = {  % define "bluecirc" circle
      draw,
      circle,
      fill=blue!10,
      minimum width=0.7cm,
    }
  ]

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[bluecirc](a) {$a$};
\draw ( 0.5, 0.5) node[bluecirc](b) {$b$};

\draw [->] (a) to[bend left=30] node[above]{0} (b);
\draw [->] (b) to[bend left=30] node[below]{1} (a);

\end{tikzpicture}
\end{center}
```

16 Automata

Here's an example of a DFA:



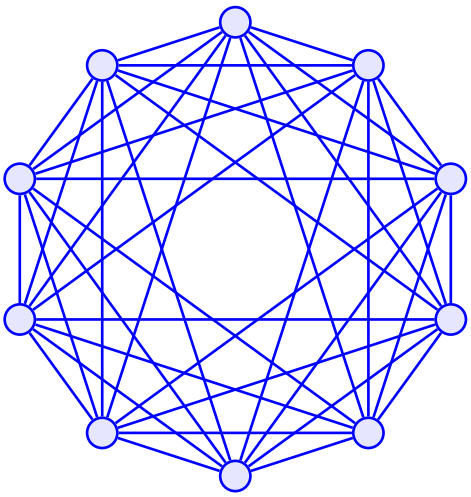
LaTeX code

```
Here's an example of a DFA:

\begin{center}
\begin{tikzpicture}[
    >=triangle 60,
    line width=0.02cm,
    state/.style = {
      draw, circle, minimum width=0.7cm,
    },
    accept/.style = {
      draw, circle, minimum width=0.7cm, double, double distance=1pt
    }
  ]

\draw[help lines] (-4,-2) grid (4,2);

\draw (-2.5, 0.5) node[state ](q0) {$q_0$};
\draw ( 0.5, 0.5) node[state ](q1) {$q_1$};
\draw ( 3.5, 0.5) node[accept](q2) {$q_2$};

\coordinate (A) at (-3.5, 0.5);

\draw [->] (A)  --                                                  (q0);
\draw [->] (q0) --                              node[below]{1}      (q1);
\draw [->] (q1) --                              node[below]{1}      (q2);
\draw [->] (q0) to[bend left=30]                node[above]{0}      (q2);
\draw [->] (q1) edge[loop below, distance=0.7cm] node[below]{0}     (q1);
```

```
\draw [->] (q2) to[bend left=60]                node[below]{0,1} (q0);

\end{tikzpicture}
\end{center}
```

17 Graph

Here's an example of an undirected graph:



Of course there are too many nodes and edges to type. So of course I wrote a program to generate the tikz code:

```
from math import cos, sin, pi

print(r'''
\begin{center}
\begin{tikzpicture}[
    >=triangle 60,
    line width=0.02cm,
    anode/.style = {
      draw=blue, circle, minimum width=0.4cm, fill=blue!10, line width=1pt
    },
    aline/.style = {
      blue, line width=1pt
    },
  ]
''')

r = 3.0
points = {}
for i,d in enumerate(range(0, 360, 36)):
    deg = d + 90
    t = deg * pi / 180.0
    point = (r * cos(t),  r * sin(t))
```

```
    point = (round(point[0], 2), round(point[1], 2))
    points[i] = point

for k,v in points.items():
    x,y = v
    print(r'\draw (%5s, %5s) node[anode](%s) {};' % (x,y,k))

keys = points.keys()
for k in keys:
    m = (k + 1) % len(keys)
    print(r'\draw [aline] (%s) -- (%s);' % (k, m))
    m = (k + 2) % len(keys)
    print(r'\draw [aline] (%s) -- (%s);' % (k, m))
    m = (k + 3) % len(keys)
    print(r'\draw [aline] (%s) -- (%s);' % (k, m))
    m = (k + 4) % len(keys)
    print(r'\draw [aline] (%s) -- (%s);' % (k, m))

print(r'''\end{tikzpicture}
\end{center}
''')
```

(By the way, is there an Eulerian cycle in the graph? What about a Hamiltonian cycle?)

LaTeX code

```
Here's an example of an undirected graph:

\begin{center}
\begin{tikzpicture}[
    >=triangle 60,
    line width=0.02cm,
    anode/.style = {
      draw=blue, circle, minimum width=0.4cm, fill=blue!10, line width=1pt
    },
    aline/.style = {
      blue, line width=1pt
    },
  ]

\draw (  0.0,   3.0) node[anode](0) {};
\draw (-1.76,  2.43) node[anode](1) {};
\draw (-2.85,  0.93) node[anode](2) {};
```

```
\draw (-2.85, -0.93) node[anode](3) {};
\draw (-1.76, -2.43) node[anode](4) {};
\draw ( -0.0,  -3.0) node[anode](5) {};
\draw ( 1.76, -2.43) node[anode](6) {};
\draw ( 2.85, -0.93) node[anode](7) {};
\draw ( 2.85,  0.93) node[anode](8) {};
\draw ( 1.76,  2.43) node[anode](9) {};
\draw [aline] (0) -- (1);
\draw [aline] (0) -- (2);
\draw [aline] (0) -- (3);
\draw [aline] (0) -- (4);
\draw [aline] (1) -- (2);
\draw [aline] (1) -- (3);
\draw [aline] (1) -- (4);
\draw [aline] (1) -- (5);
\draw [aline] (2) -- (3);
\draw [aline] (2) -- (4);
\draw [aline] (2) -- (5);
\draw [aline] (2) -- (6);
\draw [aline] (3) -- (4);
\draw [aline] (3) -- (5);
\draw [aline] (3) -- (6);
\draw [aline] (3) -- (7);
\draw [aline] (4) -- (5);
\draw [aline] (4) -- (6);
\draw [aline] (4) -- (7);
\draw [aline] (4) -- (8);
\draw [aline] (5) -- (6);
\draw [aline] (5) -- (7);
\draw [aline] (5) -- (8);
\draw [aline] (5) -- (9);
\draw [aline] (6) -- (7);
\draw [aline] (6) -- (8);
\draw [aline] (6) -- (9);
\draw [aline] (6) -- (0);
\draw [aline] (7) -- (8);
\draw [aline] (7) -- (9);
\draw [aline] (7) -- (0);
\draw [aline] (7) -- (1);
\draw [aline] (8) -- (9);
\draw [aline] (8) -- (0);
\draw [aline] (8) -- (1);
\draw [aline] (8) -- (2);
\draw [aline] (9) -- (0);
\draw [aline] (9) -- (1);
```

```
\draw [aline] (9) -- (2);
\draw [aline] (9) -- (3);
\end{tikzpicture}
\end{center}

Of course there are too many nodes and edges to type. So of course I wrote a program
to generate the tikz code:
\VerbatimInput[frame=single, fontsize=\small]{graph.py}

(By the way, is there an Eulerian cycle in the graph?
What about a Hamiltonian cycle?)
```