

```

1  import sys
2  from tkinter import *
3  class MyCanvas(Canvas):
4      def __init__(self, master, hLinewidth=1, vLinewidth=1, radius=2,
5      **kwargs):
6          Canvas.__init__(self, master, kwargs)
7          self.hLinewidth = hLinewidth
8          self.vLinewidth = vLinewidth
9          self.radius = radius
10
11      def create_segment_h(self, x, y, l):
12          self.create_line(x, y, x + l, y, width=self.hLinewidth)
13          self.create_oval(x - self.radius, y - self.radius, x + self.radius,
14          y + self.radius, fill='black')
15          self.create_oval(x + l - self.radius, y - self.radius, x + l -
16          self.radius, y + self.radius, fill='black')
17
18      def create_segment_v(self, x, y, l):#
19          self.create_line(x, y, x, y + l, width=self.vLinewidth)
20          self.create_oval(x - self.radius, y - self.radius, x + self.radius,
21          y + self.radius, fill='black')
22          self.create_oval(x - self.radius, y + l - self.radius, x +
23          self.radius, y + l + self.radius, fill='black')
24
25      def create_line_h(self, x, y, l):#
26          self.create_line(x, y, x + l, y, width=self.hLinewidth)
27
28      def create_line_v(self, x, y, l):
29          self.create_line(x, y, x, y + l, width=self.vLinewidth)
30
31  if __name__ == '__main__':
32      n = int(input('please input the number n: '))
33      winw, winH = 2400 * 0.4, 1500 * 0.4
34      hMargin, vMargin = winw // 20, winH // 20
35      hScale, vScale = (winw - 2 * hMargin) // (2*n-2), (winH - 2 * vMargin)
36      // (n - 1)
37      root = Tk()
38      root.title('A Typical Transposition Network with n=%d (Drawn by Python
39      Tkinter)' % n)
40      cvs = MyCanvas(root, bg='white', width=winw, height=winH)
41      for i in range(n):
42          cvs.create_line_h(hMargin, vMargin+i*vScale, (2*n-2)*hScale)
43      for i in range(n-1):
44          for j in range(i//2+1):
45              cvs.create_segment_v(hMargin+(i+1)*hScale, vMargin+i*vScale-
46              2*j*vScale, vScale)
47          if i!=n-2:
48              cvs.create_segment_v(winw-hMargin-
49              (i+1)*hScale, vMargin+i*vScale-2*j*vScale, vScale)
50      cvs.pack()
51      root.mainloop()

```

