```python
import sys
from tkinter import *

class MyCanvas(Canvas):
    def __init__(self, master, hLineWidth=1, vLineWidth=1, radius=2, **kwargs):
        Canvas.__init__(self, master, kwargs)
        self.hLineWidth = hLineWidth
        self.vLineWidth = vLineWidth
        self.radius = radius

    def create_segment_h(self, x, y, l):
        self.create_line(x, y, x + l, y, width=self.hLineWidth)
        self.create_oval(x - self.radius, y - self.radius, x + self.radius, y + self.radius, fill='black')
        self.create_oval(x + l - self.radius, y - self.radius, x + l - self.radius, y + self.radius, fill='black')

    def create_segment_v(self, x, y, l):#
        self.create_line(x, y, x, y + l, width=self.vLineWidth)
        self.create_oval(x - self.radius, y - self.radius, x + self.radius, y + self.radius, fill='black')
        self.create_oval(x - self.radius, y + l - self.radius, x + self.radius, y + l + self.radius, fill='black')

    def create_line_h(self, x, y, l):#
        self.create_line(x, y, x + l, y, width=self.hLineWidth)

    def create_line_v(self, x, y, l):
        self.create_line(x, y, x, y + l, width=self.vLineWidth)

if __name__ == '__main__':
    n = int(input('please input the number n: '))
    sortingNetwork = Sorter(n)

    winW, winH = 2400 * 0.4, 1500 * 0.4
    hMargin, vMargin = winW // 20, winH // 20
    hScale, vScale = (winW - 2 * hMargin) // (2*n-4), (winH - 2 * vMargin) // (n - 1)

    root = Tk()
    root.title('A Typical Transposition Network with n=%d (Drawn by Python Tkinter)' % n)
    cvs = MyCanvas(root, bg='white', width=winW, height=winH)

    for i in range(n):
        cvs.create_line_h(hMargin,vMargin+i*vScale,(2*n-4)*hScale)
    for i in range(n-1):
        print(i)
        for j in range(i//2+1):
            print(i,j)
            cvs.create_segment_v(hMargin+i*hScale,vMargin+i*vScale-2*j*vScale,vScale)
            cvs.create_segment_v(winW-hMargin-i*hScale,vMargin+i*vScale-2*j*vScale,vScale)
```

```
47
48      cvs.pack()
49      root.mainloop()
```