Colin Gasiewicz Lab 5
Put together the histogram code as a PyCharm project. **0.5%**

Currently, the keys are shown next to each other with the same space between two consecutive keys. Change the code such that the space is proportional to their distance. For example, the distance between key values 7 and 4 should be 3 times as long as that between key values 4 and 3. **0.5%**

Create two test cases with varied numbers of values. At least one of them must have at least 10 key values.

Submit a screenshot of the final histogram.

Main.py code

```
#!/usr/bin/python3
# main.py

import random as rand
from ch04_frequncy_dist_with_histogram import frequencyChart
def list_factory(n, minimum, maximum):
      """
      Makes a list of random integers of size n with integers in [min,
max].
      Requires n >= 1 and maximum > minimum.

      :param n: size of the array to be built
      :param min: min value that will appear in the array
      :param max: max value that will appear in the array
      :return:
      """
      if n >= 1:
      if maximum > minimum:
            lst = []
            for i in range(n):
                  lst.append(rand.randint(minimum, maximum))
                  print(str(lst[i]) + ', ')
            return lst
      else:
            print('maximum must be greater than minimum')
            return [1]
      else:
      print('n must be greater than or equal to 1')
      return [2]
```

```
rand.seed("1234")
maximum = 7
minimum = 0
lst1 = list_factory(10, minimum, maximum)
lst2 = list_factory(25, minimum, maximum)
frequencyChart(lst1)
frequencyChart(lst2)
```

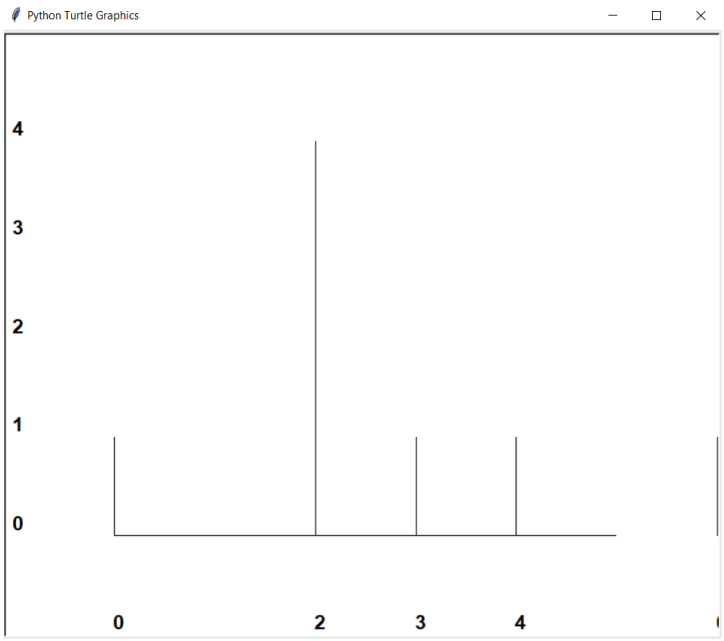
Ch04_frequncy_dist_with_histogram.py Code:

```
#!/usr/bin/python3
# Ch04_frequncy_dist_with_histogram.py

import turtle

def frequencyChart(alist):
      countDict = {}

      for item in alist:
      if item in countDict:
            countDict[item] = countDict[item] + 1
      else:
            countDict[item] = 1
      itemList = list(countDict.keys())
      minItem = 0
      maxItem = len(itemList) - 1
      itemList.sort()

      countList = countDict.values()
      maxCount = max(countList) + 1

      wn = turtle.Screen()
      chartT = turtle.Turtle()
      wn.setworldcoordinates(-1, -1, maxItem+1, maxCount)
      chartT.hideturtle()

      chartT.up()
      chartT.goto(0, 0)
      chartT.down()
      chartT.goto(maxItem, 0)
      chartT.up()
      for i in range(maxCount):
      chartT.goto(-1, i)
```

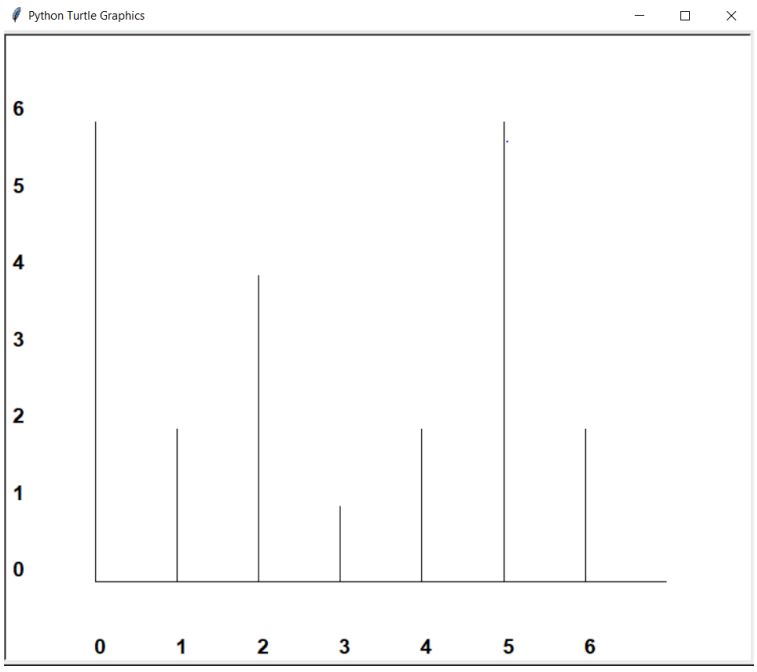
```
      chartT.write(str(i), font=("Helvetica", 16, "bold"))
      itr = 0
      for index in range(min(itemList), max(itemList)):
      chartT.goto(index, -1)
      # write the bottom number
      if index in itemList:
            chartT.write(str(itemList[itr]), font=("Helvetica", 16,
"bold"))
            chartT.goto(index, 0)
            chartT.down()
            chartT.goto(index, countDict[itemList[itr]])
            chartT.up()
            itr += 1
      else:
            chartT.goto(index, -1)
      wn.exitonclick()
```

Output Screenshots:

Test1:



Test2:


Python Turtle Graphics
6
5
4
3
2
1
0
0
1
2
3
4
5
6