```
Timer unit: 1e-06 s

Total time: 7.66835 s
File: <ipython-input-5-faf2ad6c3ad2>
Function: main at line 26

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    26                                           def main(n, alpha, T):
    27                                               """
    28                                               n: there are n points in the unit square
    29                                               alpha: alpha is the portion of points you redistribute at every time step
    30                                               T: number of iterations
    31                                               return:
    32                                               length_array: list of caluclated values L(0), L(1),\dots, L(T-1)
    33                                               plot the diagram L(t) for t= 0, 1,...,T-1
    34                                               """
    35         1        769.0    769.0      0.0      x = np.random.uniform(size=n)
    36         1         62.0     62.0      0.0      y = np.random.uniform(size=n)
    37         1        282.0    282.0      0.0      positions = list(zip(x,y))
    38         1          2.0      2.0      0.0      length_array = []
    39         1        678.0    678.0      0.0      position_status = dict((key, value) for (key, value) in zip(positions, [False]*(len(positions)
    40         1         17.0     17.0      0.0      starting_point = random.choice(positions)
    41         1         58.0     58.0      0.0      positions.remove(starting_point)
    42         1          2.0      2.0      0.0      position_status[starting_point] = True
    43
    44
    45         6         10.0      1.7      0.0      for t in range(0,T):
    46                                                   #perturbation of points
    47         5          6.0      1.2      0.0          if t > 0:
    48         4        136.0     34.0      0.0              positions = list(position_status.keys())
    49         4          8.0      2.0      0.0              perturb_no = n*alpha
    50         4     310650.0  77662.5      4.1              new_pos, old_pos = returnPerturbations(perturb_no, positions)
    51         4      74406.0  18601.5      1.0              positions = [rp for rp in positions if rp not in old_pos]
    52         4         22.0      5.5      0.0              positions.extend(new_pos)
    53         4      76328.0  19082.0      1.0              position_status = {key: position_status[key] for key in list(position_status.keys()) i
    54
    55      5005       6132.0      1.2      0.1          for p in range(0,n):
    56      5000       7644.0      1.5      0.1              if len(positions) > 1:
    57      4998       5496.0      1.1      0.1                  path_length = 0
    58      4998       5985.0      1.2      0.1                  if p == 0:
    59         5          6.0      1.2      0.0                      previous_point = starting_point
    60
    61                                                           #find shortest path through all points
    62      4998    6877684.0   1376.1     89.7                  current_point,increment = minEucDistance(previous_point,positions)
    63      4998       8920.0      1.8      0.1                  path_length += increment
    64      4998     103802.0     20.8      1.4                  positions.remove(current_point)
    65      4998       8429.0      1.7      0.1                  position_status[current_point] = True
    66      4998       5637.0      1.1      0.1                  previous_point = current_point
    67
    68         5          9.0      1.8      0.0          length_array.append(path_length)
    69
    70         1      20248.0  20248.0      0.3      plt.xlabel('Iteration (t)')
    71         1         40.0     40.0      0.0      plt.ylabel('L(t)')
    72         1        283.0    283.0      0.0      plt.title('Mypoic Distance Calculation')
    73         1       7342.0   7342.0      0.1      plt.bar([i for i in range(0,T)],length_array)
    74         1     147253.0 147253.0      1.9      plt.show()
    75
    76         1          2.0      2.0      0.0      return length_array
```