

```

source code:
import heapdict as heapdict
import math
from random import random
from random import choice
import time
from loop import loop
import numpy as np
def __init__(self, v):...
def hasOutNeighbor(self, v):...
def hasInNeighbor(self, v):...
def hasNeighbor(self, v):...
def getOutNeighbors(self):...
def getInNeighbors(self):...
def getOutNeighborsWithWeights(self):...
def getInNeighborsWithWeights(self):...
def addOutNeighbor(self, v, wt):...
def addInNeighbor(self, v, wt):...
def __init__(self):...
def addVertex(self, n):...
def addDiEdge(self, u, v, wt=1):...
def addBiEdge(self, u, v, wt=1):...
def getDirEdges(self):...
def __init__(self):...
def addVertex(self, n):...
def addDiEdge(self, u, v, wt=1):...
def addBiEdge(self, u, v, wt=1):...
def getDirEdges(self):...
def randomGraph(n, p, wts=[1]):...
def BFS(w, G):...
def BFS_shortestPaths(w, G):...
def dijkstraDumb(w, G):...
def dijkstraDumb_shortestPaths(w, G):...
def dijkstra(w, G):...
def dijkstra_shortestPaths(w, G):...
def runTrials(myFn, nVals, pFn, numTrials=1):...
def smallFrac(n):...
if __name__ == '__main__':

```

