

Sep 08, 14 22:05	csc710sbse: hw1:Theisen	Page 1/1
<div>46 ?.?..!+?.?.?.+.+.?.!+!+...?.?.+.+.?.+.+.?.+.+.?.+.+.?. +.+.?.?.?.!+..... 98!+..... 99?.+.+.?.+.+.+.+.+.?.+.+.+.+.+.?.+.+.?.+.+. 99 ?.+.?.+.+.?.+. 99+. 99 99 99 ?.+.+.+.+.+.+.+.+.?.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+. 99 +. 99!+..... 99 Found best - s: 0.934735340081 e: 0.00593315090177</div>		

Sep 08, 14 22:10

csc710sbse: hw1:Theisen

Page 1/1

```

#From Class Discussion 8/26/2014
from __future__ import division
import sys, re, random, math
sys.dont_write_bytecode = True

kmax = 500
cooling = .6

#From Baseline Study - schaffer_trials.py
emax = 44
emin = 0

rand = random.random

#Structure from SA Lecture
def say(x):
    sys.stdout.write(str(x)); sys.stdout.flush()

def Energy(x):
    rawAns = math.fabs((x*x) + (x-2)*(x-2))
    ans = (rawAns + emin) / (emax - emin)
    return ans

def Neighbor(x):
    return random.uniform(-10, 10)

#Structure from SA Lecture
def main():
    s = random.uniform(-10, 10) #random start
    e = Energy(s)
    sBest = s
    eBest = e
    k = 1
    say(int(math.fabs(eBest-1)*100))
    say(' ')
    while k < kmax:
        sNew = Neighbor(s)
        eNew = Energy(sNew)
        if eNew < eBest:
            sBest = sNew
            eBest = eNew
            say('!!')

        myRand = random.random()

        if eNew < e:
            s = sNew
            e = eNew
            say('++')
        #Probability Check from SA Lecture
        elif math.exp(-1*(eNew-e)/(k/kmax**cooling)) < myRand:
            #P function should be between 0 and 1
            #more random hops early, then decreasing as time goes on
            s = sNew
            e = eNew
            say('??')
            #print 'Random Hop! (?)'

        say('.')
        k = k + 1
        if k % 50 == 0 ^ k != kmax:
            print ' ',
            say(int(math.fabs(eBest-1)*100))
            say(' ')

    print '\nFound best - s:', sBest, ' e:', eBest

main()

```

Sep 01, 14 12:53

csc710sbse: hw1:Theisen

Page 1/1

```

#From Class Discussion 8/26/2014
from __future__ import division
import sys, re, random, math
sys.dont_write_bytecode = True

kmax = 100000 #100000 run trial for max

rand = random.random

low = 100

def say(x):
    sys.stdout.write(str(x)); sys.stdout.flush()

def Energy(x):
    ans = math.fabs((x*x) - (x-2)*(x-2))
    return ans

def Neighbor(x):
    return random.uniform(-10, 10)

def eMax():
    s = random.uniform(-10, 10) #random start
    e = Energy(s)
    sBest = s
    eBest = e
    k = 1
    while k < kmax:
        sNew = Neighbor(s)
        eNew = Energy(sNew)
        if eNew > eBest: #find largest difference
            sBest = sNew
            eBest = eNew
        k = k + 1

    print 'Found eMax - s:', sBest, ' e: ', eBest

def eMin():
    s = random.uniform(-10, 10) #random start
    e = Energy(s)
    sBest = s
    eBest = e
    k = 1
    while k < kmax:
        sNew = Neighbor(s)
        eNew = Energy(sNew)
        if eNew < eBest: #find smallest difference
            sBest = sNew
            eBest = eNew
        k = k + 1

    print 'Found eMin - s:', sBest, ' e: ', eBest

eMax()
eMin()

```