Luddes portkodssekvens

En teknolog som glömt sin fyrsiffriga portkod tryckte sej igenom alla tiotusen kombinationer så här.

00000001000200030004000500060007000800090010011...9999

Det kräver fyrtiotusen tryckningar. Men man kan klara sej med bara tiotusentre tryckningar om man har en supersmart sekvens där varje fyrsiffrigt tal förekommer någonstans. Hur ser sekvensen ut?

Problemträdets urförälder 0000 har tio barn 00000, 00001,..., 00009, varav det första är ett dumbarn. Breddenförst eller djupetförst? Vi vet att trädet har djupet tiotusen och att alla lösningar är lika långa, därför går djupetförst bra. Men breddenförst skulle kräva biljoner poster!

```
import sys

sys.setrecursionlimit(20000)
# too high gives segmentation fault

size = 10000  # 100, 1000, etc
visited = [False] * size
numVisited = 0
calls = 0

def visit(currentSeries, nextDigit):
    global visited, numVisited, calls

    calls += 1
    newNum = currentSeries % (size/10) * 10 + nextDigit
    if visited[newNum]: return

    visited[newNum] = True
    numVisited += 1
    newSeries = currentSeries * 10 + nextDigit

    if numVisited == size:
        print newSeries
        print 'after', calls, 'calls to visit.'
        sys.exit()

    for i in range(10):
        visit(newSeries, i)

    visited[newNum] = False
    numVisited -= 1


def start(currentSeries, nextDigit):
    print 'Starting at', currentSeries, nextDigit,
    print 'leads to'
    visit(currentSeries, nextDigit)

start(999,9)
```

```
size = 100

Starting at 0 0 leads to
(00)10203040506070809112131415161718192232425262728293343536373839444546474849556
5758596676869778798899 0
after 640 calls to visit.
```

(Eftersom vi använder tal "försvinner" de inledande nollorna i utskriften.)

```
Starting at 1 1 leads to
1100120213031404150516061707180819092232425262728293343536373839445464748495565 7
585966768697787988991
after 629 calls to visit.
```

```
Starting at 9 9 leads to
9900102030405060708091121314151617181922324252627282933435363738394454647484955 6
57585966768697787988 9
after 541 calls to visit.
```

```
size = 10000

Starting at 999 9 leads to
...
after 54991 calls to visit.
```

LABYRINT

En svartvit grafikskärm kan beskrivas som en boolesk
1000x1000−matris, där true betyder en svart bildpunkt.
På skärmen finns en labyrint. Ge en algoritm som finner
en väg från övre vänstra till nedre högra hörnet och som
bara går till vågrätt och lodrätt liggande vita grannpunkter.

Lösning

Problemträdet har (0,0) som stamfar och tillåtna grannpunkter
ger söner. I en dumsonsmatris markerar man vilka punkter
man redan besökt. Breddenförst fungerar då bra och ger
kortaste vägen.

```
# coding:iso−8859−1

from queue import Queue
from sys import exit

q=Queue()
M=10
N=10

def makesons(koordinatpar):
  (xx,yy)=koordinatpar
  if xx==M−1 and yy==N−1: print "Ute!"
  for dx in [−1,0,1]:
    for dy in [−1,0,1]:
      x=xx+dx; y=yy+dy
      if 0<=x<M and 0<=y<N and ok[x][y]:
        ok[x][y]=False
        q.put((x,y))

karta=open("karta").readlines()
ok=range(M)
for i in range(M):
  ok[i]=[]
  for j in range(N):
    if karta[i][j]==" ":
        ok[i].append(True)
    else:
        ok[i].append(False)


q.put((0,0))
while not q.isempty():
    makesons(q.get())
```

Men djupet först är lättare att tänka sej, rent praktiskt (man knyter
fast ett snöre i startpunkten). Om det finns möjlighet att gå i cirklar
måste man markera var man varit, precis som i lösningen ovan.

HÅLL REDA PÅ MEDIA (Tildatenta 030308)

Under gulfkriget var det väldigt svårt för armestaben att hålla reda
på alla TV−bolag som for omkring och rapporterade i öknen. För att
hålla reda på dem användes en hashvektor. Koden fungerade inte som
avsett och man har nu gett i uppdrag åt en f.d. tildastudent att titta
på en misstänkt del av koden:

```
from string import find

p=100;
hashvektor = [0]*p
alfabet="abcdefghijklmnopqrstuvxyz"

def put(tvbolagsnamn, tvbolag):
    hashcode = 0
    for i in range(len(tvbolagsnamn)):
        alfanum = find(alfabet, tvbolagsnamn[i])+1
        hashcode += alfanum
    hashcode = hashcode % p
    hashvektor[hashcode] = tvbolag
```

Vad är det för fel på koden? Beskriv hur man kan förbättra den.
Namnen på TV−bolagen kan antas bestå av högst tre bokstäver.
Det kommer inte att förekomma mer än 75 TV−bolag.

NIX TILL TELEFONFÖRSÄLJNING (TILDA−tenta 000603)

Föreningen för konsumentskydd vid marknadsföring per telefon har
startat ett register dit den som inte vill bli uppringd av
telefonförsäljare kan anmäla sig.

Till att börja kommer kontrollen att ske genom att företaget sänder
sin telefonlista till nix och får tillbaka en lista där de nixade
numren markerats.

Vilka av följande metoder kan föreningen använda sig av? Vilken är bäst?
Binärträd, bloomfilter, hashtabell.

Ett framtida mål är att kontroll också skall kunna ske över internet.
Då måste kontrollen ske snabbt men man vill också försäkra sig om att
ingen ska kunna få ut en lista över alla nixade telefonnummer.

Vilken metod passar bäst för internet−kontrollen?