

1 STRYKORD

```
0 # coding: iso-8859-10
```

Ordet orkan är ett strykord eftersom man kan stryka sista bokstaven om och om igen och bara få riktiga ord. orkan - orka - ork - or - o Uppgiften är att finna det längsta strykordet i svenska språket. Ordlistan finns på fil. Rita problemträdet och jämför olika tänkbara algoritmer.

Rekursiv tanke:

1. Uppdatera rekord om ord längre än rekord
2. hitta söner till ord+a
3. hitta söner till ord+b
4. ...,
5. hitta söner till ord+ö

Basfall:

Om ordet inte finns return

Add current directory to python path

```
29 import sys, os
30
31 sys.path.append( os.getcwd() )
32
33
34 def byggut(ord):
35     global record
36     if len(ord)>len(record): record=ord
37     for tkn in alfabet:
38         if svenska.exists(ord+tkn):
39             byggut(ord+tkn)
40
41 from bintree import Bintree
42 alfabet="abcdefghijklmnopqrstuvwxyzåäö "
43 record=""
44
45 print 'Ord: ',
```

Ord:

```
46 svenska=Bintree()
47 svenskfil = open("words.txt")
48 for rad in svenskfil.readlines():
49     print rad.strip(),
50     svenska.put(rad.strip()) # Strippa returtecknet
```

o hur ska ett **or** utan ork orka äta ost i en orkan undrar jag

```
53
54 byggut("")
55
56 print
57
58 print
59
60 print "Lägsta strykord:", record
```

Lägsta strykord: orkan

2 SJUOR TILL HUNDRA

```
0 # coding:iso-8859-1
```

Det gäller att skriva talet 100 med enbart sjuor och dom fyra räknesätten, till exempel så här:

$$7 + 7 / 7 * 7 * 7 = 98$$

Det var ett gott försök som inte nådde ända fram. För att man ska få använda / måste divisionen gå jämnt ut. Rita problemträdet och diskutera bästa algoritm för att avgöra OM problemet är lösbart.

Om man dessutom vill veta hur lösningen ser ut krävs en mer komplicerad datastruktur. Beskriv den och skissa ett program.

Add current directory to python path

```
21 import sys , os
22
23 sys.path.append( os.getcwd() )
24
25 from queue import Queue
26 from bintree import Bintree
27
28 q=Queue()
29 dumsons=Bintree()
30
31
32 def makesons(tal):
33     if tal==100:
34         print "Hundra!"
35         return True
36     if not dumsons.exists(tal):
37         dumsons.put(tal)
38
39         q.put(tal+7)
40         q.put(tal-7)
41         q.put(tal*7)
42         if (tal%7==0): q.put(tal/7)
43         return False
44
45
46 q.put(0)
47 bool=False
48 while not q.isempty():
49     bool=makesons(q.get())
50     if bool: break
```

Hundra!

2 SJUOR TILL HUNDRA version 2

```
0 # coding:iso-8859-1
```

Det gäller att skriva talet 100 med enbart sjuor och dom fyra räknesätten, till exempel så här.

$$7 +7 /7 *7 *7 =98$$

Det var ett gott försök som inte nådde ända fram. För att man ska få använda / måste divisionen gå jämnt ut. Rita problemträdet och diskutera bästa algoritm för att avgöra OM problemet är lösbart.

Om man dessutom vill veta hur lösningen ser ut krävs en mer komplicerad datastruktur. Beskriv den och skissa ett program.

Add current directory to python path

```
21 import sys, os
22 sys.path.append( os.getcwd() )
23 from queue import Queue
24 from bintree import Bintree
25
26 q=Queue()
27 dumsons=Bintree()
28
29 class Node:
30     tal=0
31     far=None
32     op=''
33
34 def makesons(far):
35     tal=far.tal
36     if tal==100:
37         writechain(far)
38         return True
39     if not dumsons.exists(tal):
40         dumsons.put(tal)
41
42         insert(tal+7, '+7', far)
43         insert(tal-7, '-7', far)
44         insert(tal*7, '*7', far)
45         if (tal%7==0): insert(tal/7, '/7', far)
46     return False
47
48 def insert(tal, op, far):
49     nod=Node()
50     nod.tal=tal
51     nod.op = op
52     nod.far=far
53     q.put(nod)
54
55 def writechain(p):
56     if p is None: return
57     writechain(p.far)
58     print '(%s) '%(p.op), p.tal ,
59
60 q.put(Node())
61 bool=False
62 while not q.empty():
63     bool=makesons(q.get())
64     if bool:
65         break
```

() 0 (+7) 7 (+7) 14 (*7) 98 (*7) 686 (+7) 693 (+7) 700 (/7) 100

3 LABYRINT

```
0 # coding: latin
```

En svartvit grafiskskärm 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.

Add current directory to python path

```
13 import sys, os
14 sys.path.append( os.getcwd() )
15
16 from queue import Queue
17 from sys import exit
18
19 q=Queue()
20 M=10
21 N=10
22
23 def makesons(koordinatpar):
24     xx, yy = koordinatpar
25     dist=matris[xx][yy]
26     if xx==M-1 and yy==N-1:
27         for ny_rad in matris:
28             for pos in ny_rad:
29                 print str(pos) + ' '* (2-len(str(pos))),
30             print
31         print
32         print "Ute!"
33         return True
34
35 for dx, dy in [(0,1),(0,-1),(1,0),(-1,0)]:
36     x=xx+dx; y=yy+dy
37     if 0<=x<M and 0<=y<N and 0 == matris[x][y] and matris[x][y] is not 'x':
38         :
39         matris[x][y]= 1 + dist
40         q.put((x,y))
41
42 return False
43
44 karta=open("karta").readlines()
45 matris=[]
46 for rad in karta:
47     ny_rad=[]
48     for i in range( len( rad ) ):
49         if rad[i]==' ':
50             ny_rad.append(0)
51         elif rad[i]=='x':
52             ny_rad.append('x')
53     matris.append(ny_rad)
54
55
56
57
58
59
60
61
62
63
```

```
64 print 'Startmatrix'
```

Startmatrix

```
68 for ny_rad in matris:
69     for pos in ny_rad:
70         print str(pos) + ' '* (2-len(str(pos))),
71         print
```

```

0  x  x  x  x  x  x  x  x  x
0  0  0  0  0  0  0  0  0  x
x  x  x  x  0  x  x  0  x  x
x  0  0  0  0  x  0  0  x  x
x  0  0  x  x  x  x  0  0  x
x  0  0  0  0  0  x  0  0  0
x  x  x  x  0  0  x  0  0  x
x  0  0  0  0  x  x  x  0  x
x  0  0  x  x  x  x  x  0  x
x  x  x  x  x  x  x  x  0  0

```

```
69 print
```

```
70 print 'Slutmatrix'
```

Slutmatrix

```

72
73 matris[0][0]=1
74 ok=range(M)
75 for i in range(M):
76     ok[i]=[]
77     for j in range(N):
78         if karta[i][j]==" ":
79             ok[i].append(True)
80         else:
81             ok[i].append(False)
82
83
84 q.put((0,0))
85 bool=False
86 while not q.isempty():
87     bool=makesons(q.get())
88     if bool:
89         break

```

```

1  x  x  x  x  x  x  x  x  x
2  3  4  5  6  7  8  9  10 x
x  x  x  x  7  x  x  10 x  x
x  11 10 9  8  x  12 11 x  x
x  12 11 x  x  x  x  12 13 x
x  13 12 13 14 15 x  13 14 15
x  x  x  x  15 16 x  14 15 x
x  19 18 17 16 x  x  x  16 x
x  0  19 x  x  x  x  x  17 x
x  x  x  x  x  x  x  x  18 19

```

Ute!