Háskóli Íslands

Greining Reiknirita TÖL403

Verkefni 1

Erling Óskar Kristjánsson

Kennari:
Hjálmtýr Hafsteinsson

14. febrúar 2019

Efnisyfirlit

1 Verkefnalýsing - Forritunarverkefni **1**

2 Spurningar & Svör **2**
Gráðugt Reiknirit . . . 2
Kvik Bestun . . . 3
Forrit og samanburður á niðustöðum . . . 4

1 Verkefnalýsing - Forritunarverkefni

- Sölumaður að fara í langferð. Hægt er að hugsa ferð hans sem ferðalag út eftir jákvæðu talnalínunni og að hann byrji á staðsetningu 0. Á leiðinni eru n hótel á stöðum $a_1 < a_2 < ... < a_n$, þar sem a_i er fjöldi kílómetra frá upphafstaðnum.

- Sölumaðurinn vill helst ferðast 200 km á dag. Hann verður að gista á einhverju þessara n hótela og enda á síðasta hótelinu (sem er í fjarlægð a_n).

- Nú eru hótelin ekki endilega 200 km frá hvert öðru, þannig að ef sölumaðurinn ferðast x km á tilteknum degi þá er kostnaður fyrir þann dag $(200-x)^2$. Það skiptir því ekki máli hvort sölumaðurinn þarf að fara 10 km of langt eða 10 km of stutt, kostnaðurinn er 100. **Markmiðið er að lágmarka heildarkostnað ferðarinnar.**

Innlegg: Ferðakostnaður er semsagt 0 ef sölumaðurinn ferðast akkúrat 200 km á dag. Samanlagður ferðakostnaður verður væntanlega hærri ef hann ferðast alltaf 190km frekar en 210km, því hann þyrfti þá að ferðast í fleiri daga.

2 Spurningar & Svör

(a) Sýnið einfalt gráðugt reiknirit fyrir verkefnið. Sýnið einnig með mótdæmi að það finnur ekki alltaf bestu lausn.

Lausn:

Skilgreinum fylki `A[1..n]` $= [a_1, a_2, ..., a_n]$ þar sem a_i er fjarlægð hótels með heiltöluvísi i frá upphafspunktinum $a_0 = 0$ og skilyrðin eru eins og í verkefnalýsingunni að ofan.

Ákvörðunin sem sölumaður þarf að taka fyrir sérhvert hótel er þ.l.s. einföld: *Á hann að gista þar eða ekki?* Það sem hefur áhrif á ákvörðunina er hvort það sé lægri kostnaður við að gista á þessu hóteli, því næsta, eða einhverju af þeim sem eftir eru.

Með þessum upplýsingum og `A` sem víðværa breytu skilgreinum við gráðugt reiknirit sem reynir einungis að besta næsta stopp, þar sem $i < j$ og sölumaðurinn er staddur á hótelinu með vísi i:

```
A[0] = 0    // Upphafspunkturinn
gradKostn = 0
gradSkref = 0
i = gradSkref
while i < n:
    gradSkref++
    xi = A[gradSkref]-A[i]
    ki = (200 - xi)^2
    for j in range(i+2, n+1):
        xij = A[j]-A[i]
        kij = (200 - xij)^2
        if kij <= ki:   // Skrefið er betra en það sem við áður héldum að væri best
            gradSkref = j   // Tökum þetta skref ef ekkert betra finnst
            ki = kij        // Byrjum að bera saman við þetta nýja skref
    gradKostn += ki
    i = gradSkref
return gradKostn
```

Þetta reiknirit finnur hins vegar ekki alltaf bestu lausn. Skoðum eftirfarandi hótelstaðsetningar:

```
A[0..3] = [0, 215, 351, 480]
Reikniritið myndi velja að stoppa á:
    A[1] = 215       (200-215)^2 =  225
    A[2]       (200-(351-215))^2 = 4096
    A[3]       (200-(480-351))^2 = 5041
    Heildarkostnaður             = 9362

Hins vegar hefði verið betra að fórna smá í öðru skrefi og stoppa á:
    A[1] = 215       (200-215)^2 =  225
    A[3]       (200-(480-215))^2 = 4225
    Heildarkostnaður             = 4450
```

Almennt ef kostnaðurinn við að fara til `A[j]` frá `A[i]` er bara aðeins lægri en til `A[j+1]` frá `A[i]`, þá tekur gráðuga reikniritið (líklega) vitlausa ákvörðun, eins og í þessu tilfelli.

Þess vegna skulum við leita að endurkvæmri formúlu sem finnur betri lausn, og varpa henni svo yfir í kvikt bestunarverkefni.

(b) Sýnið hvernig hægt er að leysa verkefnið með kvikri bestun. Setjið fram endurkvæma formúlu og útskýrið svo hvernig hægt er að komast hjá því að reikna sömu gildin aftur með því að geyma milliniðurstöður.

Lausn:

Skilgreinum fylki `A[1..n]` $= [a_1, a_2, ..., a_n]$ þar sem a_i er fjarlægð hótels með heiltöluvísi i frá upphafspunktinum $a_0 = 0$ og skilyrðin eru eins og í verkefnalýsingunni að ofan.

Ákvörðunin sem sölumaður þarf að taka fyrir sérhvert hótel er þ.l.s. einföld: *Á hann að gista þar eða ekki?* Það sem hefur áhrif á ákvörðunina er hvort það sé lægri kostnaður við að gista á þessu hóteli eða næsta, þegar litið er á heildarferðina.

Með þessum upplýsingum og `A` sem víðværa breytu, þá skilgreinum við eftirfarandi endurkvæma formúlu, þar sem $i < j$ og sölumaðurinn er staddur á hótelinu með vísi i:

$$Kostnadur(i,j) = \begin{cases} (200 - (A[j] - A[i])^2) & \text{ef } j = n \\ min\left\{ \begin{array}{l} Kostnadur(i, j+1), \\ \left(200 - (A[j] - A[i])^2\right) + Kostnadur(j, j+1) \end{array} \right\} & \text{annars} \end{cases}$$

Ef $j = n$ þá er sölumaðurinn kominn á síðasta hótelið og verður að stöðva þar.
Annars tekur sölumaðurinn þá ákvörðun sem kostar minna:
- að stoppa ekki á hóteli númer j, heldur skoða $j + 1$ frá i
- að stoppa og færa sig yfir á hótel númer j, og skoða næst $j + 1$ frá j

Til að breyta verkefninu í Kvika Bestun þá viljum við geyma lágmarkskostnaðinn við að komast á hótel númer $1, 2, ..., j$, t.d. í fylki `mK[1..j]`, og þá verður lágmarkskostnaðurinn við að komast á hótel númer $j + 1$ einfaldlega

$$\min_{i=1,2,...,j} \left\{ \left(200 - (A[j+1] - A[i])^2\right) + mK[i] \right\}$$

Þessi almenna lausn virkar fyrir margar kostnaðarformúlur (þ.e.a.s aðrar en $(200-x)^2$), og margar mismunandi dreifingar af fjarlægðum milli hótela.

Við gætum geymt og ítrað yfir færri lágmarkskostnaði af síðustu hótelum `mK` ef við erum viss um að dreifingin sé á ákveðnu bili, eins og í tilfellinu þar sem lengdin milli hótelanna er $X \sim Unf(100, 300)$. Sú lausn myndi ekki virka fyrir allar dreifingar, því t.d. ef kostnaðarformúlan og fjöldi hótela er 100 eins og í okkar tilfelli, en hótelin væru öll á 2ja km millibili, þá væri lágmarkskostnaðurinn sá að stoppa einungis á síðasta hótelinu (og við þyrftum að ítra yfir öll fyrri hótel til að ganga úr skugga um það).

(c) Forritið bæði reikniritin og berið niðurstöður þeirra saman á slembigögnum sem þið búið til, sem eru þ.a. $n = 100$ og fjarlægð í næsta hótel $X \sim Unf(100, 300)$, og berið saman við aðra t.d. Jafndreifð með stærra bil, Normaldreifð með meðaltal í kringum 200, eða veldisdreifð með meðaltal í kringum 200.

Lausn:

```
# -*- coding: utf-8 -*-
"""
Namskeid:   Greining Reiknirita tol403
Kennari:    Hjalmtyr Hafsteinsson
Stofnun:    Haskoli Islands
Höfundur:   Erling Oskar Kristjansson

------------------------------------------------------------------------------

NOTKUNARLYSING:
v1.py er lausn a Verkefni 1 i ofangreindu namskeidi.
Keyra skal forritid sem eina heild, thvi thad notast vid vidvaerar breytur sem
annars gaetu ruglast nema notandi þekki adferdirnar og breyturnar ur forritinu.
Forritid prentar videigandi nidurstodur.

  Adferdin fDreifdHotel(..) framleidir og skilar auka tuple sem er
ekki notad vid utreikninga, en getur verid gagnlegt og frodlegt ad glugga i.

"""
import random
import numpy as np

''''''''''''''''''''''''''Adferdir vid mat a kostnadi''''''''''''''''''''''''''

''''''''''''''' Hjalparfoll '''''''''''''''
''' Skilar fjarlaegd milli hotela numer i og j '''
fjarlaegd = lambda i,j : np.abs((hotelfylki[j] - hotelfylki[i]))
''' Skilar kostnadinum vid ad ferdast fjarlaegd x '''
kostnadur = lambda x : (200 - x)**2

'''
Notkun  h = fDreifdHotel(aa,bb,nn,f)
Fyrir:  nn er heiltala
        f er tviundaradgerd sem skilar tolu
        aa og bb eru breytur af gerd sem f raedur vid
Eftir:  h er tuple med tveimur tuples
        h[0] eru nn tolur framkalladar med f(aa,bb)
        h[1] eru nn+1 tolur, nefnilega
    h[1] = [0, h[0][0], h[0][0]+h[0][1], ..., h[0][0]+...+h[0][nn-1]]
'''
def fDreifdHotel(aa,bb,nn,f):
    AA = [0]
    BB = []
    for i in range(nn):
        BB.append(f(aa,bb))
        AA.append(AA[i]+BB[i])
    return tuple(BB),tuple(AA)
```

```python
''' "Gráðuga reikniritið á bara að reyna að besta næsta stopp,
        og ekki að horfa neitt fram í tímann."
Gradugt reiknirit(adferd) sem gerir akkurat thad:
  Itrar yfir moguleg hotel og velur besta skrefid i hverju skrefi. '''
def kostnadurGradugt():
    gradKostn = 0
    gradSkref = 0
    i = gradSkref
    while i < hotelfjoldi:
        gradSkref += 1
        ki = kostnadur(fjarlaegd(i,gradSkref))
        for j in range(i+2, stadafjoldi):
            kij = kostnadur(fjarlaegd(i,j))
            if kij <= ki:
                gradSkref = j
                ki = kij
        gradKostn += ki
        i = gradSkref
    return gradKostn



'''
Adferd sem notar kvika bestun til ad reikna laegsta heildarkostnad ferdarinnar.
Notkun:  kb = KostnadurKB()
Eftir:   kb er tuple sem inniheldur laegsta kostnadinn til midad vid ad
         serhvert hotel a bilini [1,hotelfjoldi] geti verid sidasta hotelid.
             kb[hotelfjoldi] inniheldur tvi laegsta heildarkostnad ferdarinnar
'''
def KostnadurKB():
    minKostn = [0]
    k1 = kostnadur(fjarlaegd(0,1))
    minKostn.append(k1)
    for j in range(2,stadafjoldi):
        kj = kostnadur(fjarlaegd(j-1,j)) + minKostn[j-1]
        for i in range(j-2,-1,-1):     # fra j-2 nidri 0
            kij = kostnadur(fjarlaegd(i,j)) + minKostn[i]
            kj = min(kij,kj)
        minKostn.append(kj)
    return tuple(minKostn)
```

```
'''''''''''''''  V1 - Gerd gagna  ''''''''''''''
unf = lambda a,b : random.uniform(a,b)
nmork, emork        = 100, 300
hotelfjoldi         = 100
stadafjoldi         = hotelfjoldi+1
hotel               = fDreifdHotel(nmork,emork,hotelfjoldi,unf)
hotelfjarlaegdir    = hotel[0]
hotelfylki          = hotel[1]
''''''''''''''''''''''''''''''''''''''''''''''''

kostnadurKB0 = KostnadurKB()
kostnadurGradugt0 = kostnadurGradugt()

print('Nidurstada med', hotelfjoldi,
    'hotel thar sem fjarlaegd a milli hotela er jafndeifd slembitala a milli',
    nmork, 'og', emork, ':')
print('Heildarkostnadur ferdar kvik bestun:', kostnadurKB0[hotelfjoldi])
print('Heildarkostnadur ferdar gradugt:', kostnadurGradugt0)
print()



'''''''''''''''  V1 - Önnur Gögn ''''''''''''''
exp = lambda a,b : random.expovariate(a/b)
mean                = 200
hotelfjoldi         = 100
stadafjoldi         = hotelfjoldi+1
hotel               = fDreifdHotel(1,mean,hotelfjoldi,exp)
hotelfjarlaegdir    = hotel[0]
hotelfylki          = hotel[1]
''''''''''''''''''''''''''''''''''''''''''''''''

kostnadurKB2 = KostnadurKB()
kostnadurGradugt2 = kostnadurGradugt()

print('Nidurstada med 100 hotel thar sem fjarlaegd a milli hotela',
    'er veldisdreifd slembitala med medaltal 200 :')
print('Heildarkostnadur ferdar kvik bestun:', kostnadurKB2[hotelfjoldi])
print('Heildarkostnadur ferdar gradugt:', kostnadurGradugt2)
print()
```

Dæmi um úttak:

```
Nidurstada med 100 hotel thar sem fjarlaegd a milli hotela
er jafndeifd slembitala a milli 100 og 300 :
Heildarkostnadur ferdar kvik bestun: 265610.7103720648
Heildarkostnadur ferdar gradugt: 272069.3282337942

Nidurstada med 100 hotel thar sem fjarlaegd a milli hotela
er veldisdreifd slembitala med medaltal 200 :
Heildarkostnadur ferdar kvik bestun: 1289751.0134447396
Heildarkostnadur ferdar gradugt: 1348741.851350025
```