Dünaamiline planeerimine

Margus Niitsoo

Mis täna toimub

- Tänane töö korraldus
  - Saame tuttavaks
  - Korda n korda:
    - Mina räägin
    - Praktika
  - Kokkuvõte
    - Üldistavad mõtted

Saame tuttavaks

- Tere, mina olen Margus
  - Keskkoolis käisin korra IOI-l
  - PhD arvutiteaduses
  - 3 aastat õppejõud TÜ-s
  - 3 aastat programmerija
    - Enda firma nimega MatchMySound
- Keeled:
  - C, JavaScript, Python, SQL, R, HaXe

Saame tuttavaks

- Kes teie olete?
  - Kaua olete programmeerinud?
  - Mis keeles?
  - Kas “ruutkeerukus” või $O(n^2)$ ütleb midagi?

Teie omavahel

- “Paaris programmeerimine”
  - Ehk ma jagan teid paaridesse
- Miks?
  - Kahekesi lihtsam aru saada,
    millest te aru ei saa, ja julgem küsida!
  - Õpetamine aitab enda teadmistes auke täita
- Tutvumiseks 3+3 minutit:
  - Räägi raskeimast ülesandest,
    mida sa lahendanud oled?

Mis täna toimub

- Tänased ülesanded
  - Müntidega summa maksmine
  - Kahe sõna omavaheline “kaugus”
  - Antud ristsummaga arvude arv
  - Koiliblikas

Sissejuhatus
Dünaamilisse planeerimisse

Sissejuhatuseks DP-sse

- Fibonacci arvude arvutamine
  - Algus 0, 1, 1, 2, 3, 5, 8, ...
  - Iga järgmine kahe eelmise summa

Rekursiivne lahendus

```
def FibR(n):
    if n == 0: return 0
    if n == 1: return 1
    return FibR(n - 1) + FibR(n - 2)
```

Mitu väljakutset N=5 korral?
N = 20 korral? N=40 korral?

Tsükliga lahendus

```
def FibT(n):
    Fib = [0]*(n+1)
    Fib[1] = 1
    for i in range(2,n+1):
      Fib[i] = Fib[i-1] + Fib[i-2]
    return Fib[n]
```

Miks see paremini töötab?

Küsimus mida küsida:

- Kui ma oskaks lahendada kõiki “väiksemaid” ülesandeid, siis...
- Kas ma saaks nende lahendustest praeguse lahenduse kokku panna?

Müntidega maksimine

- Antud müntide väärtused
  - Nt: 1,2,5,10,20,50,100,200 senti
  - Või: 1,3,5,7,11,13,17,19,23,29 senti
- Leia minimaalne müntide arv et tasuda N senti

- Mis on väiksemad ülesanded?
- Kuidas ma neid kasutada saan?

DP müntidega

```
Myndid = [1,3,5,7,11,13,17,19,23,29]

Mins = [0]
for i in range(1,N+1):
  Best = N
  for m in Myndid:
    if i-m>=0 and Mins[i-m]+1 < Best:
      Best = Mins[i-m]+1
  Mins.append(Best)

return Mins[N]
```

Lahenduse taastamine

- Kui on vaja teada, mis müntidega see miinimum juhtub?

Hoia lahendused alles

```
Myndid = [1,3,5,7,11,13,17,19,23,29]
Mins,Lah = [0],[[]]
for i in range(1,N+1):
  Parim,PL = N,[1]*N
  for m in Myndid:
    if i-m>=0 and Mins[i-m]+1 < Parim:
      Parim,PL = Mins[i-m]+1,Lah[i-m]+[m]
  Mins.append(Parim); Lah.append(PL)

return Lah[N]
```

Lahenduse taastamine

- Kui on vaja teada, mis müntidega see miinimum juhtub?
- Hoia lahendused alles
- Aga kui see liiga palju ruumi võtab?

Säilita vaid viimane samm

```
Myndid = [1,3,5,7,11,13,17,19,23,29]
Mins,LahV = [0],[0]
for i in range(1,N+1):
   Parim,PLV = N, 1
   for m in Myndid:
      if i-m>=0 and Mins[i-m]+1 < Parim:
         Parim,PLV = Mins[i-m]+1, m
   Mins.append(Parim); LahV.append(PLV)
```

.. ja taasta nende järgi

```
Myndid = [1,3,5,7,11,13,17,19,23,29]
Mins,LahV = [0],[0]
for i in range(1,N+1):
   Parim,PLV = N, 1
   for m in Myndid:
      if i-m>=0 and Mins[i-m]+1 < Parim:
         Parim,PLV = Mins[i-m]+1, m
   Mins.append(Parim); LahV.append(PLV)
```

```
j, LahN = N,[]
while(j>0):
    LahN.prepend(LahV[j])
    j-=LahV[j]
return LahN
```

Miks “planeerimine”?

- Ingl. Kl “Dynamic programming”
  - “Programming” ajastul kus arvutid veel väga levinud polnud
    - 1940ad, Richard Bellman, RAND
  - Analoogne Lineaarsele planeerimisele
    - 1940ad, Leonid Kantorovich, sai selle eest 1975 Nobeli majanduspreemia
  - “Planeerimine” sest mõlemat kasutati peamiselt tootmise ja sõjategevuse planeerimisel

Harjutamise aeg

- Lihtsad variatsioonid:
  - Kui summat ei saa maksta? (Nt [5,6],N=14)
  - Kui lahendust vaja kasvavas järjekorras?
- Sarnane ülesanne:
  - Mitte vähim müntide arv, vaid:

    Mitmel eri viisil saab?
    - (Maksmisel järjestus loeb)

Planeerimine tabelitega

Sõnade “kaugus”

- Antud kaks “sõna”
  - Kalasaba ja Salajane
- Kustutame mõlemast tähti, kuni mõlemast on alles samad tähed
  - Kalasaba <=> Salajane
  - Mitu kustutamist vaja?

Sõnade “kaugus”

- Antud kaks “sõna”
  - Kalasaba ja Salajane
- Kustutame mõlemast tähti, kuni mõlemast on alles samad tähed
  - Kalasaba <=> Salajane
- Mitu kustutamist vaja?

- Mis on väiksemad ülesanded?
- Kuidas ma neid kasutada saan?

Lahenduse taastamine

- Endiselt:
  - Märgi üles, mis oli eelmine alamlahendus, mida kasutati

Harjutamise aeg

- Edit distance: tegevused on:
  - Tähe kustutamine
  - Tähe lisamine
  - Ühe tähe muutmine
- Edasiarendus: muutuse hind:
  - Kustutamine ja lisamine: tähe jk. number tähestikus
  - Muutmine – jk. n. vahe.

Harjutamise aeg

- Tagasi müntide juurde:
  - Mitmel eri viisil on võimalik maksta summa N, kus erinevaks loeme maksmisviisi vaid juhul kui üleantud mündid on erinevad
  - s.o. järjestus ei loe

Harjutamise aeg

- Keerulisem näide
  - Naturaalarvu ristsummaks nimetatakse selle numbrite summat. Näiteks arvu 123 ristsumma on 1 + 2 + 3 = 6. Kirjutada programm, mis leiab, kui palju on selliseid antud arvust väiksemaid naturaalarve, mille ristsumma on võrdne antud arvu enda ristsummaga. (EIO 2013 lahtine v.)
  - Vihje: siin nõuab väiksematest lahendustest suurema kokku kombineerimine ka tsüklit.

Memoiseerimine

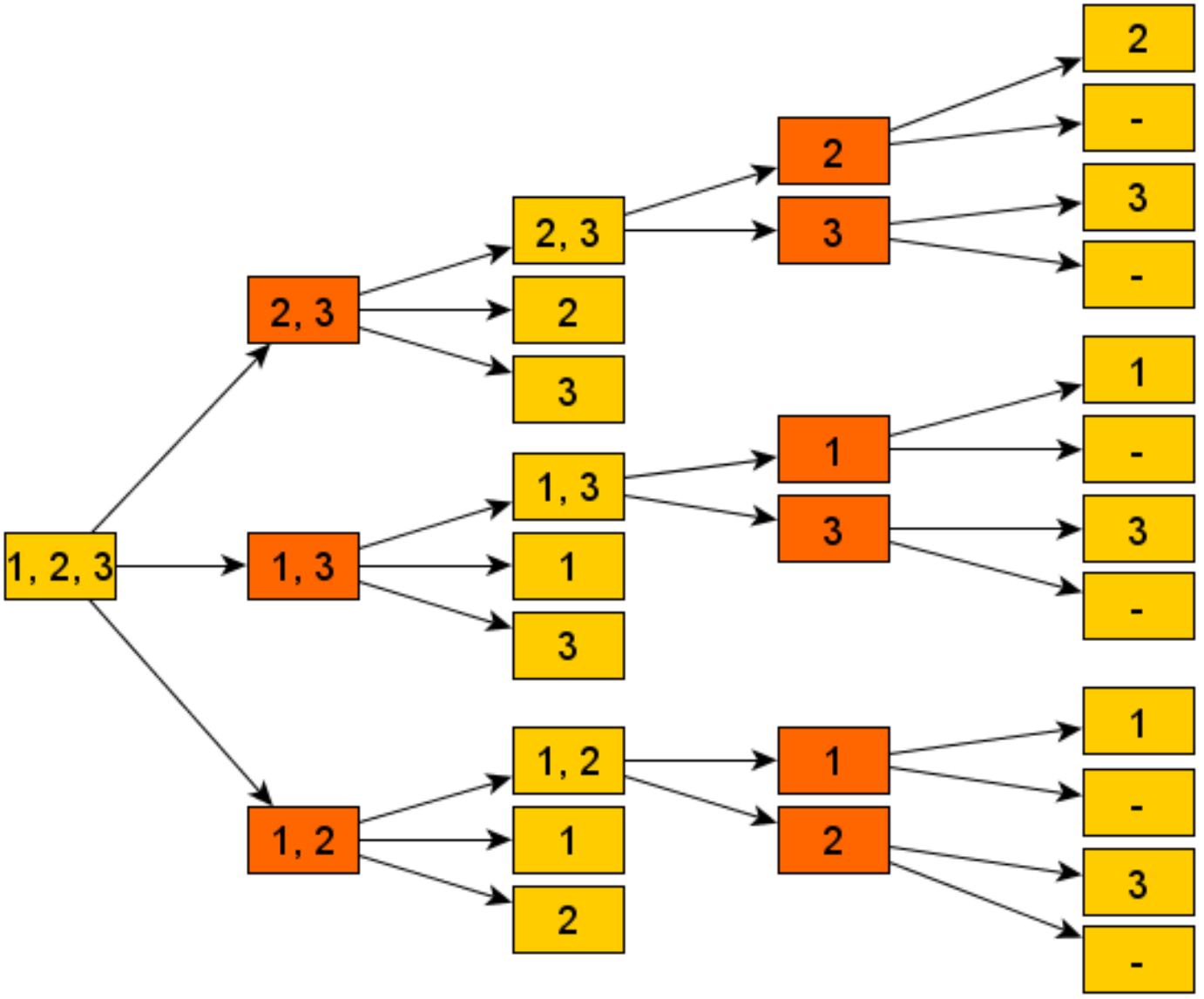
Memoiseerimine ehk laisa inimese DP

Moedisainer ja koiliblikas

- Moekunstnik Märdil on kapis N ilusat salli, millega ta käib laupäeviti moekunstnike koosolekul. Igal sallil on oma moeväärtus ja koosolekul saab Märt vastava hulga feimi. Sama salliga kohale tulla oleks suur faux pas ja Märt ei tee seda kunagi. Kantud sallid paneb ta tagasi, aga rohkem neid ei kanna.
- Märdi kapis elab ka koiliblikas Kärt, kes sööb igal pühapäeval ühele sallile augu sisse. Auguga salli enam kanda ei saa. Kärt moeväärtusest ei hooli, vaid sööb salle juhuslikult. Mingile sallile augu söömise tõenäosus on võrdeline salli pikkusega. Kärt võib sama salli süüa ka mitu korda.
- On selge, et Märt saaks koosolekul käia ülimalt N korda, kui Kärt sööks ainult juba kantud salle. Kui Kärt sööb mõnikord ka kandmata salle, jääb koosolekute arv sellevõrra väiksemaks.
- Kirjutada programm, mis leiab, kui palju Märt keskmiselt feimi saab, kui ta kasutab parimat strateegiat, aga Kärt sööb salle juhuslikult. Tegevus algab nädala alguses, seega esimese koosoleku ajaks on kõik sallid veel terved

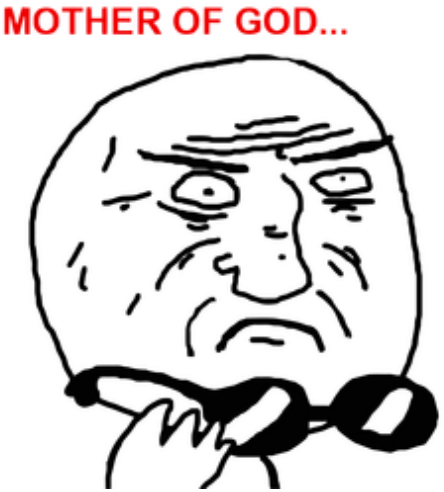
Järgnevad Targo Tennisbergi slaidid (sest ma olen laisk)

Koi - läbivaatus



Koi – keerukuse analüüs

- N Märdi käiku, seejärel N Kärdi käiku
  - Primitiivse lahenduse keerukus on $(N!)^2$
- Max N=20
  - $(20!)^2 = 5{,}919*10^{36}$
  - Arvestades 10^8 läbivaatust sekundis, kulub $2*10^{21}$ aastat



Natuke nagu Fibonacci?

- Väga paljud harud korduvad!
  - Sest ei loe, kuidas kapi seisuni jõuti, ainult kapi seis ise...
- Erinevaid olukordi palju vähem kui (20!)^2
  - Ainult 2^20 ehk ~1 000 000

Tagasi Fibonacci juurde

```
def FibR(n):
    if n == 0: return 0
    if n == 1: return 1

    Fn = FibR(n - 1) + FibR(n - 2)
    return Fn
```

Memoiseerimine

```
F = {}
def FibM(n):
    if n == 0: return 0
    if n == 1: return 1
    if n in F: return F[n]
    F[n] = FibM(n - 1) + FibM(n - 2)
    return F[n]
```

Ehk jätame vahetulemused lihtsalt meelde

Natuke nagu Fibonacci?

- Väga paljud harud korduvad!
  - Sest ei loe, kuidas kapi seisuni jõuti, ainult kapi seis ise...
- Erinevaid olukordi vaid 2^n
  - Niiet läbivaatus, aga pidage vahetulemused meeles
    - (bitimaadlemine)

Harjutamise aeg

- Lihtsam variant:
  - Võtke suvaline varasem lahendatud ülesanne, ja tehke see rekursiivselt ja memoisatsiooniga ümber
- Raskem variant:
  - Proovige koiliblika ül. lahendus valmis teha

Kokkuvõte

Mis täna toimus

- Müntidega summa maksmine
  - Kõige lihtsam ühes suunas DP
  - Lahenduse taastamine
- Kahe sõna omavaheline “kaugus”
  - DP korraga kahel väärtusel (“tabel”)
- Koiliblikas
  - Memoiseerimine ja 2^n

Kuidas ära tunda

- Variantide läbivaatus jääks aeglaseks
- Suurem ülesanne taandub väiksemate peale
- Loeb vaid hetke olukord, mitte sinna jõudmise teekond
  - Koiliblika näide
  - Müntide loendamine ver 2

Hägune hierarhia

Kiirus

Paindlikkus

Ahned algoritmid

Dünaamiline planeerimine

Variantide läbivaatus

Kuhu edasi

- Harjutage, harjutage, harjutage
  - UVA: Volume 5: Dynamic prog.
  - Project Euler – valdav enamus on DP

Tänan kuulamast!

Küsimused on teretulnud!