

Variantide läbivaatus



Ahto Truu

ahto.truu@ut.ee



Kombinatorikaülesanne

Objektide kombineerimise võimaluste

- loetlemine
- loendamine
- summeerimine
- minimaalse/maksimaalse leidmine



Näide: maksmine

Mitmel moel saame maksta summa S senti, kui meil on 1-, 2-, 5-, 10-, 20- ja 50-sendised mündid?

- kui kõiki münte on piiramatult?
- kui neid on vastavalt n_1, n_2, n_5, \dots ?
- mida teha, kui müntide nominaalid pole ette fikseeritud?



Näide: bitiülesanne

- genereerida kõik N-bitised jadad
- kui palju neid kokku on?
 - 2^N
- kas järjekord on oluline?
 - isegi kui pole, peame mingi valima
 - näiteks 000, 001, ..., 111



Iteratsioonimeetod

Algoritm

- loome esimese variandi
- kuni on variant:
 - kasutame/loendame variandi v
 - leiame järgmise variandi



Tagurdusmeetod (1)

Algoritm OTSI(v)

- kui v on täielik variant:
 - kasutame/loendame variandi v
- muidu:
 - leiame täiendusvariantide hulga W
 - iga $w \in W$ jaoks:
 - OTSI($v + w$)



Tagurdusmeetod (2)

Globaalne v

Algoritm OTSI

- kui v on täielik variant:
 - kasutame/loendame variandi v
- muidu:
 - leiame täiendusvariantide hulga W
 - iga $w \in W$ jaoks:
 - $v := v + w$
 - OTSI
 - $v := v - w$



Näide: lipuülesanne

- NxN malelaud
- paigutada sellele N lippu
- ükski ei tohi olla ühegi teise tule all

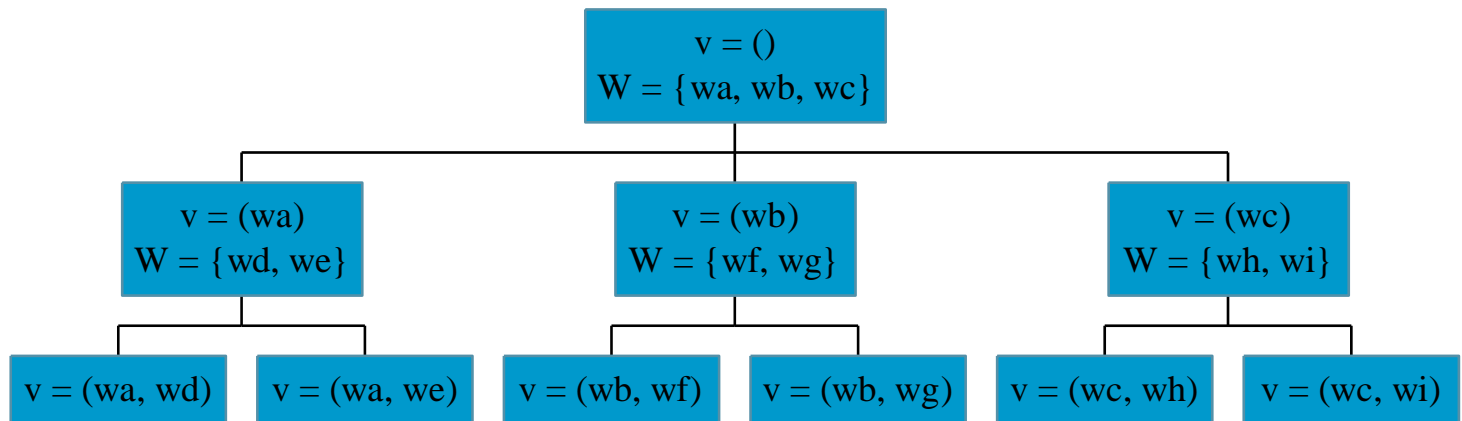
- lipp tulistab:
 - piki rida vasakule ja paremale
 - piki veergu üles ja alla
 - piki diagonaale 4 suunas



Lipud: tagurdusmeetod

- **Jooksev variant:**
 - juba paigutatud lippude loetelu
- **Täiendamisvõimaluste hulk:**
 - malelaua väljade loetelu
- **Variandi täiendamine:**
 - uuele lipule koha valimine

Otsingupuu





Otsingupuu suurus

Parameetrid

- puu sügavus n
- puu hargnevus m
 - igal sammul m -elemendiline W
 - igal tipul m alluvat

Puu suurus ja töömaht

- kokku puus m^n lehte
- tehtav töö $\Omega(m^n)$



Hoiatav näide: male

- Keskmiselt 38 käiku seisus
- Hea arvuti:
 - vaatleb sekundis 1 000 000 seisu
 - vaatab sekundiga 4 käiku ette
- Väga hea arvuti:
 - vaatleb sekundis 100 000 000 seisu
 - vaatab sekundiga 5 käiku ette
- Superarvuti:
 - vaatleb sekundis 10 000 000 000 seisu
 - vaatab sekundiga 6 käiku ette



Hoiatav näide: lipud

- **Naiivne lahendus:**
 - N lippu → puu sügavus N
 - $N \times N$ ruutu → igal sammul $\sim N^2$ varianti
 - jäme hinnang $(N^2)^N$
- **3x3 laud: 729**
- **5x5 laud: 9765625**
- **7x7 laud: 678223072849**



Olekuruumi ahendamine

- Vältida sama seisu korduvat vaatlemist
 - algseis S, käigid A, B; $S' = S+A+B$
 - algseis S, käigid C, D; $S'' = S+C+D$
 - kui $S' = S''$, siis on raiskamine või isegi vale
- Lihtsamatel juhtudel saab kordusi vältida
- Keerulisematel juhtudel tuleb neid ära tunda



Lipud: ahendamine

- Veergude vaatlemine vasakult paremale:
 - N lippu \rightarrow puu sügavus N
 - N rida \rightarrow i. sammul N varianti
 - jäme hinnang N^N
- 3x3 laud: 27 (27x vähem)
- 5x5 laud: 3125 (3125x vähem)
- 7x7 laud: 823543 (823543x vähem)



Puu pügamine

- **Tunda halb seis ära võimalikult vara**
 - jätta see kohe kõrvale
 - väldime terve alampuu töötlemist
- **Liigne agarus on ohtlik**
 - võime kaotada otsitava lahenduse



Lipud: pügamine

- Ainult vabad ruudud jooksvas veerus:
 - N lippu \rightarrow puu sügavus N
 - N rida \rightarrow i. sammul $\sim(N-i)$ varianti
 - jäme hinnang $N!$
- 3x3 laud: 6 (4,5x vähem)
- 5x5 laud: 120 (26x vähem)
- 7x7 laud: 5040 (163x vähem)



Lipud: optimeerimine (1)

Eelmiste lippudega võrdlemine:

- vabaduse kontroll $O(N)$
- lipu lisamine/eemaldamine $O(1)$

Ruutude olekute märkimine:

- vabaduse kontroll $O(1)$
- lipu lisamine/eemaldamine $O(N)$



Lipud: optimeerimine (2)

Veergude/ridade/diagonaalide olekud

- veerud: ilmutamata kujul
 - read: rida[r]
 - tõusvad diagonaalid: $\text{diag1}[r-v]$
 - langevad diagonaalid: $\text{diag2}[r+v]$
- vabaduse kontroll $O(1)$
 - lisamine/eemaldamine $O(1)$
 - kokkuhoid N -kordne



Harjutamiseks

- CMSis 3 ülesannet
- igaüks annab 5 häkkeripunkti
- CMS avatud esmasp-pühapä