Dünaamiline Programmeerimine

Oliver-Matis Lill

November 4, 2017

Fibonacci numbrid

- Fibonacci numbrid on defineeritud järgmiselt:

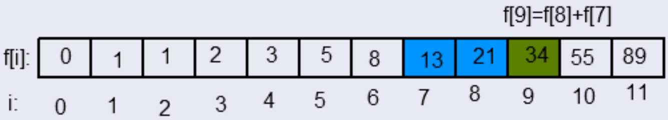
$$f_0 = 0; \quad f_1 = 1; \quad f_i = f_{i-1} + f_{i-2}$$

- Arvuta Fibonacci numbrite väärtused kuni 10^5 numbrini:

$$f_0, f_1, \dots, f_{10^5-1}, f_{10^5}$$

- Kuidas te seda lahendaksite?

Fibonacci numbrid

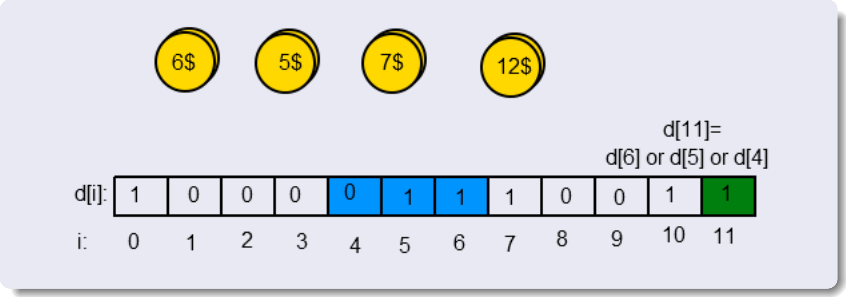


Mündid

- Riik annab välja N mündutüüpi, kus müntide väärtused rahaühikutes on $d_1, d_2, \ldots, d_N$
- Leia kas antud müntidega on võimalik maksta C rahaühikut
- Kõik väärtused on täisarvud ning kehtivad järgmised tingimused

$$C \leq 10^4; \quad N \leq 100$$

Mündid



Seljakott

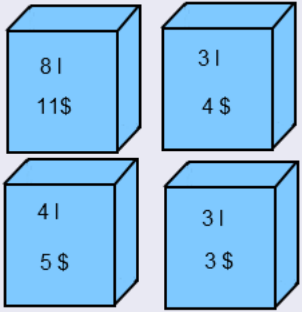
- Sul on N eset, kus igal esemel on mingi kaal w_i ja väärtus v_i
- Sul on seljakott kuhu sa võid asetada asju summaarse kaaluga kuni W
- Leia mis on maksimaalse võimaliku seljakotti asetatava esemetekomplekti väärtuse
- Sisendis on kõik numbrid täisarvud ja kehtivad järgmised piirid:

$$W \leq 10^5; \quad N \leq 100$$

Seljakott

i \ j	0	1	2	3	4
10	0	11	11	11	12
9	0	11	11	11	11
8	0	11	11	11	11
7	0	0	4	9	9
6	0	0	4	5	7
5	0	0	4	5	5
4	0	0	4	5	5
3	0	0	4	4	4
2	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

d[10][4] =
max(d[10][3], d[7][3] + 3)



DP üldisemalt

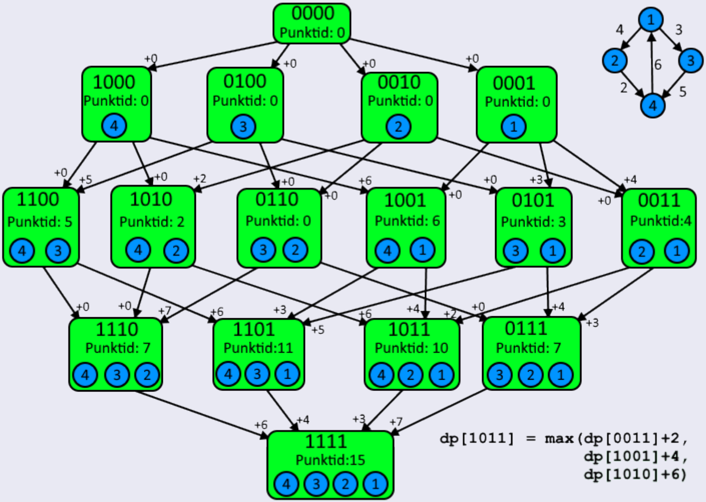
- Sarnane matemaatilisele induktsioonile
- Tüüpiline protseduur on järgmine
  1. Jaota probleem mingiks hulgaks alamprobleemideks
  2. Leia valem millega saab "suurema" alamprobleemi arvutada "väiksemate" alamprobleemide tulemuste abil
  3. Rakenda valemit et lahendada iga alamprobleem ja lõpuks ka esialgne probleem
- Dünaamiline planeerimise kasutamiseks läheb vaja alamprobleemideks jaotamise skeemi ja rekursioonivalemit mis kirjeldab alamprobleemide vahelisi seoseid
- Samal probleemil võib olla mitu skeemi ja rekursioonivalemit.
- Kui skeem loob $O(S)$ alamprobleemi ja rekursioonivalemi arvutamiseks tuleb teha $O(R)$ tehet, siis keerukus tuleb $O(SR)$

Bitmask-i skeem

- Oletame et meil on vaja arvutatda midagi elementide hulga peal
- Tegemist on alamprobleemideks jaotamise skeemiga kus alamprobleemid vastavad alamhulkadele
- Siin saab nn "bitmask"-e kasutada alamhulkade tähistamisel. Hulgal $S = \{x_1, x_2, x_3, x_4, x_5\}$ bitmask 01101 tähistab alamhulka $S_1 = \{x_2, x_3, x_5\}$
- Tüüpiliselt taandab algoritmi keerukuse $O(n!)$ pealt $O(2^n)$ või mingi sarnase keerukuse peale.

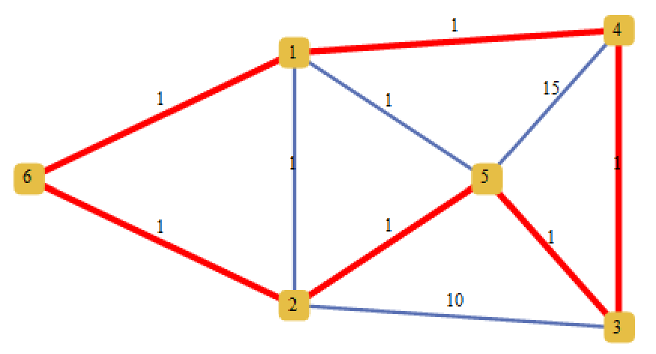
- Sul on N objekti
- Sa soovid need objektid järjekorda panna
- Sul on M reeglit kus iga reegel on kujul: "Kui objekt i on enne objekti j, siis sa saad p punkti"
- $N \leq 20$
- Mis on maksimaalne võimalik punktisumma mille sa võid järjestamisega teenida

Järjestamine




```
dp[1011] = max(dp[0011]+2,
               dp[1001]+4,
               dp[1010]+6)
```

- Meil on *n* linna ja linnade vahel on teatud pikkustega teed.
- Leia lühim ringkäik mis läbib kõik linnad täpselt üks kord ja jõuab alguslinna tagasi.
- Näide Rändkaupmehe ringkäigust:



Travelling Salesman probleem

- Idee: See kuidas me edasi liikuda saame sõltub ainult sellest mis linnad me oleme varem läbinud ja kus linnas me praegu oleme.
- Skeem:
  $D_S^{(i)}$ =minimaalse teekonna pikkus millega on võimalik läbida hulgas S olevad linnad ja jõuda lõpuks linna i
- Rekursioonivalem:

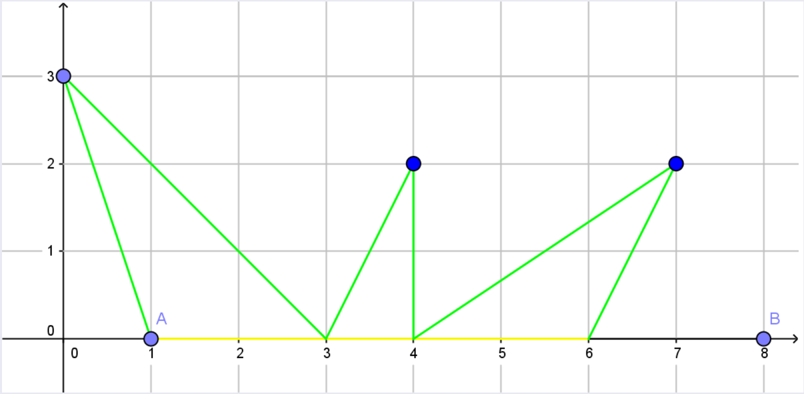
$$D_S^{(i)} = \min\{D_{S\setminus\{i\}}^{(j)} + d_{j,i} | j \in S \setminus \{i\}\}$$

  Kus $d_{j,i}$ on tee pikkus linnast j linna i.
- Meil on $O(2^n)$ seisundit ja rekursioonivalemi saab arvutada keerukusega $O(n^2)$, seega meil on $O(n^2 2^n)$ algoritm.

Ülesanne "Bear and Floodlight"

- n tulvarit tasandil, neid saab keerata ja igaüks valgustab mingit nurka a_i
- Karu soovib liikuda sirgjooneliselt punktist $(l, 0)$ punkti $(r, 0)$ mööda valgustatud teed.
- Suuna tulvarid nii, et karu saaks valgustatud teed mööda võimalikult lähedale oma sihtkohale kõndida.

Näide



Ülesanne "Bear and Floodlight"

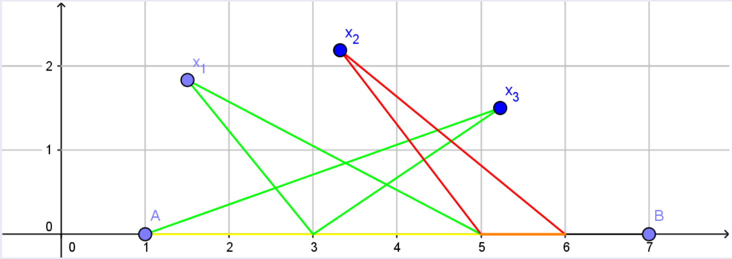
- Idee: Leia iga tulvarite alamhulga jaoks, kui kaugele on nendega võimalik kõndida
- Skeem:
  $D_S =$ maksimaalne kaugus kuhu tulvarite hulgaga S on võimalik kõndida.
- Rekursioonivalem:

$$D_S = \max\{\mathrm{ext}(D_{S\setminus\{x\}}, x) | x \in S\}$$

Kus $\mathrm{ext}(d, x)$ on saadud valgustatud teekonna pikkus, kui me valgustame pikkusega d valgustatud piirkonda tulariga x edasi. On lihtne näha et x peaks olema suunatud nii, et selle valgustatud lõigu vasak serv oleks $(l + d, 0)$.

Ülesanne "Bear and Floodlight"

- $D_{\{x_1,x_2,x_3\}} = \max(\mathrm{ext}(D_{\{x_1,x_2\}}, x_3), \mathrm{ext}(D_{\{x_1,x_3\}}, x_2), \mathrm{ext}(D_{\{x_2,x_3\}}, x_1))$
- Näide $\mathrm{ext}(D_{\{x_1,x_3\}}, x_2)$ vastavale valikule:



Kokkuvõte

- DP on tõenäoliselt kõige levinum probleemilahendamise meetod võistlusprogrammeerimise valdkonnas üldse
- Tihti osa suuremast ülesannetest
- Raskemate ülesannete puhul võib rekursioonivalem väga keerukaks minna
- Probleemil võib olla mitu erinevat rekursioonivalemit ja alamprobleemide skeemi, mõned neist paremad kui teised.
- Sobiva reukrsioonivalemi ja skeemi tuvastamine võib raske olla