Dinamično programiranje

Nino Bašić

6. maj 2011

				7				
			3		8			
		8		1		0		
	2		7		4		4	
4		5		2		6		5

- pot:
  - začetek: 1. vrstica, konec: zadnja vrstica
  - korak: {dol + levo, dol + desno}
- Q: Kakšna je največja možna vsota po neki taki poti?

				7				
			3		8			
		8		1		0		
	2		7		4		4	
4		5		2		6		5

- pot:
  - začetek: 1. vrstica, konec: zadnja vrstica
  - korak: {dol + levo, dol + desno}
- Q: Kakšna je največja možna vsota po neki taki poti?
- Preblisk: Vedno stopim na večjo!

$$
\begin{array}{ccccccccc}
 & & & & 7 & & & & \\
 & & & 3 & & 8 & & & \\
 & & 8 & & 1 & & 0 & & \\
 & 2 & & 7 & & 4 & & 4 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
$$

- pot:
  - začetek: 1. vrstica, konec: zadnja vrstica
  - korak: {dol + levo, dol + desno}
- Q: Kakšna je največja možna vsota po neki taki poti?
- Preblisk: Vedno stopim na večjo! Fail!

Zgled #1

Rekurzivno preglejmo vse možne poti

Ena možnost:

$$
\begin{array}{ccccccccc}
 & & & & 7 & & & & \\
 & & & 3 & & 8 & & & \\
 & & 8 & & 1 & & 0 & & \\
 & 2 & & 7 & & 4 & & 4 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
\qquad \Sigma = 17
$$

Zgled #1

Rekurzivno preglejmo vse možne poti

Ena možnost:

$$
\begin{array}{ccccccccc}
 & & & & 7 & & & & \\
 & & & 3 & & 8 & & & \\
 & & 8 & & 1 & & 0 & & \\
 & 2 & & 7 & & 4 & & 4 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
\qquad \Sigma = 17
$$

- n ... višina trikotnika
- naredimo $n - 1$ korakov
- na vsakem koraku imamo 2 možnosti
- Skupno število poti: 2^{n-1}

Ena možnost:

7
3 8
8 1 0
2 7 4 4
4 5 2 6 5

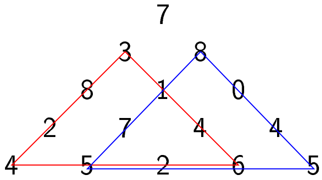
$\Sigma = 17$

- n . . . višina trikotnika
- naredimo $n - 1$ korakov
- na vsakem koraku imamo 2 možnosti
- Skupno število poti: 2^{n-1}
- Časovna zahtevnosti: $O(2^{n-1})$

Zgled #1

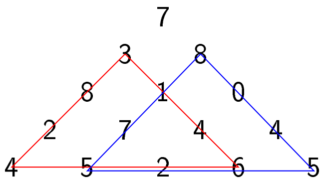
Problem razdelilmo na 2 manjša problema

1. Problem razdelimo na dva manjša podproblema iste sorte:



2. Poiščemo največjo vsoto v obeh podproblemih. (Z rekurzijo.)
3. Končna rešitev: 7 + boljša od rešitev obeh podproblemov

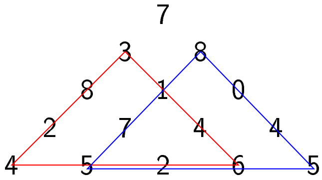
1. Problem razdelimo na dva manjša podproblema iste sorte:



2. Poiščemo največjo vsoto v obeh podproblemih. (Z rekurzijo.)
3. Končna rešitev: 7 + boljša od rešitev obeh podproblemov

Kaj opazimo? Manjše podprobleme rešujemo večkrat!

1. Problem razdelimo na dva manjša podproblema iste sorte:



2. Poiščemo največjo vsoto v obeh podproblemih. (Z rekurzijo.)
3. Končna rešitev: 7 + boljša od rešitev obeh podproblemov

Kaj opazimo? Manjše podprobleme rešujemo večkrat!

Dinamično programiranje: Isti podproblem rešuj samo enkrat!

- od spodaj navzgor
- rešimo probleme velikosti 1 (trivialno)
- ko rešujemo probleme velikosti k imamo že pripravljene rešitve vseh podproblemov velikosti $k - 1$

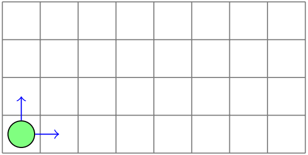
$$
\begin{array}{ccccccccc}
 & & & & 7+23 & & & & \\
 & & & 3+20 & & 8+13 & & & \\
 & & 8+12 & & 1+12 & & 0+10 & & \\
 & 2+5 & & 7+5 & & 4+6 & & 4+6 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
$$

- od spodaj navzgor
- rešimo probleme velikosti 1 (trivialno)
- ko rešujemo probleme velikosti k imamo že pripravljene rešitve vseh podproblemov velikosti $k-1$

$$
\begin{array}{ccccccccc}
 & & & & 7+23 & & & & \\
 & & & 3+20 & & 8+13 & & & \\
 & & 8+12 & & 1+12 & & 0+10 & & \\
 & 2+5 & & 7+5 & & 4+6 & & 4+6 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
$$

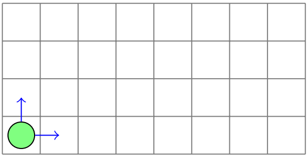
- Časovna zahtevnost: $O(n^2)$

- šahovnica velikosti $n \times m$



- v polju $(1, 1)$ je figurica [spodaj levo]
- dovoljeni premiki: {en korak v desno, en korak navzgor}
- figurico moramo pripeljati na polje (n, m) [zgoraj desno]
- Q: Na koliko različnih načinov lahko to dosežemo?

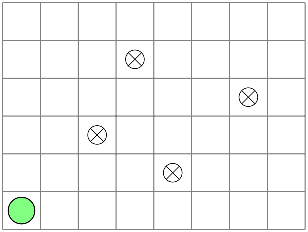
- šahovnica velikosti $n \times m$



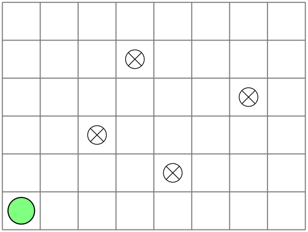
- v polju $(1, 1)$ je figurica [spodaj levo]
- dovoljeni premiki: {en korak v desno, en korak navzgor}
- figurico moramo pripeljati na polje (n, m) [zgoraj desno]
- Q: Na koliko različnih načinov lahko to dosežemo?

Izkušeni kombinatoriki poznajo odgovor: $\dfrac{(n+m)!}{n!m!}$

- Kaj pa, če nekatera polja vsebujejo $\otimes$ (mine)?

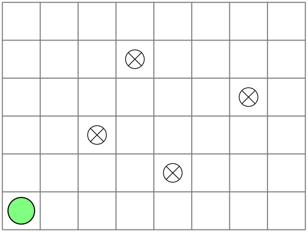


- Kaj pa, če nekatera polja vsebujejo $\otimes$ (mine)?



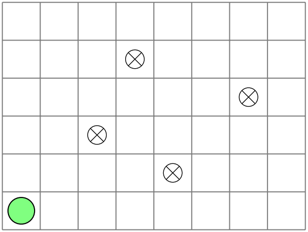
- Kombinatoriki: “Pravilo vključitev in izključitev!”

- Kaj pa, če nekatera polja vsebujejo $\otimes$ (mine)?



- Kombinatoriki: "Pravilo vključitev in izključitev!"
- Veliko število min $\implies$ komplikacije!

- Kaj pa, če nekatera polja vsebujejo $\otimes$ (mine)?



- Kombinatoriki: “Pravilo vključitev in izključitev!”
- Veliko število min $\implies$ komplikacije!
- Lahko uberemo drugačen pristop: dinamično programiranje!

Problem:

- dana sta niza a in b
- poišči dolžino najdaljšega skupnega podzaporedja

Zgled:

- $a =$ “skokica” $b =$ “tekočina”
- skupno podzaporedje: “kia”
- dolžina(“kia”) $= 3$

Problem:

- dana sta niza a in b
- poišči dolžino najdaljšega skupnega podzaporedja

Zgled:

- $a =$ "skokica" $b =$ "tekočina"
- skupno podzaporedje: "kia"
- dolžina("kia") $= 3$
- Je to najdaljše skupno podzaporedje?

Rekurzivna formula:

$$\mathrm{LCS}(a, b) = \begin{cases} 0; & \mathrm{len}(a) = 0 \\ 0; & \mathrm{len}(b) = 0 \\ \mathrm{LCS}(a[1:], b[1:]) + 1; & a[0] = b[0] \\ \max\{\mathrm{LCS}(a[1:], b), \mathrm{LCS}(a, b[1:])\}; & a[0] \neq b[0] \end{cases}$$

Premislimo, da zgornja formula res drži.

Rekurzivna formula:

$$\mathrm{LCS}(a,b) = \begin{cases} 0; & \mathrm{len}(a) = 0 \\ 0; & \mathrm{len}(b) = 0 \\ \mathrm{LCS}(a[1:], b[1:]) + 1; & a[0] = b[0] \\ \max\{\mathrm{LCS}(a[1:], b), \mathrm{LCS}(a, b[1:])\}; & a[0] \neq b[0] \end{cases}$$

Premislimo, da zgornja formula res drži.

Časovna zahtevnost (groba ocena): $O(2^{m+n})$, kjer sta m in n dolžini nizov a in b

Rekurzivna formula:

$$\mathrm{LCS}(a, b) = \begin{cases} 0; & \mathrm{len}(a) = 0 \\ 0; & \mathrm{len}(b) = 0 \\ \mathrm{LCS}(a[1:], b[1:]) + 1; & a[0] = b[0] \\ \max\{\mathrm{LCS}(a[1:], b), \mathrm{LCS}(a, b[1:])\}; & a[0] \neq b[0] \end{cases}$$

Premislimo, da zgornja formula res drži.

Časovna zahtevnost (groba ocena): $O(2^{m+n})$, kjer sta m in n dolžini nizov a in b

Bi šlo mogoče dinamično? Zakaj?

	s	k	o	k	i	c	a	\0
t								0
e								0
k								0
o							1	0
č						1	1	0
i					2	1	1	0
n					1	1	1	0
a				1	1	1	1	0
\0	0	0	0	0	0	0	0	0

	s	k	o	k	i	c	a	\0
t								0
e								0
k								0
o							1	0
č						1	1	0
i					2	1	1	0
n					1	1	1	0
a				1	1	1	1	0
\0	0	0	0	0	0	0	0	0

Časovna zahtevnost: $O(mn)$

Dan je nek izraz, ki vsebuje n celih števil in $n-1$ operatorjev (seštevanje ali množenje), npr.

$$2 + 4 * (-8) + 9 * 2 + (-5) * 7 + 9$$

V tam izrazu lahko operacije izvedeš v poljubnem vrstnem redu (oz. lahko v izraz vstaviš oklepaje).

Kolikšna je največja možna vrednost, ki jo lahko dobiš?

$$2 + 4 * (-8) + 9 * 2 + (-5) * 7 + 9$$

1. Če izberem neko operacijo (na voljo imam $(n-1)!$ izbir) in jo izračunam, dobim krajši izraz istega tipa:
   - $8 * (-8) + 9 * 2 + (-5) * 7 + 9$
   - $2 + (-32) + 9 * 2 + (-5) * 7 + 9$
   - $2 + 4 * (-1) * 2 + (-5) * 7 + 9$
   - ...

$$2 + 4 * (-8) + 9 * 2 + (-5) * 7 + 9$$

1. Če izberem neko operacijo (na voljo imam $(n-1)!$ izbir) in jo izračunam, dobim krajši izraz istega tipa:
   - $8 * (-8) + 9 * 2 + (-5) * 7 + 9$
   - $2 + (-32) + 9 * 2 + (-5) * 7 + 9$
   - $2 + 4 * (-1) * 2 + (-5) * 7 + 9$
   - ...
2. Rekurzivno rešim vse te podprobleme.

$$2 + 4 * (-8) + 9 * 2 + (-5) * 7 + 9$$

1. Če izberem neko operacijo (na voljo imam $(n-1)!$ izbir) in jo izračunam, dobim krajši izraz istega tipa:
    - $8 * (-8) + 9 * 2 + (-5) * 7 + 9$
    - $2 + (-32) + 9 * 2 + (-5) * 7 + 9$
    - $2 + 4 * (-1) * 2 + (-5) * 7 + 9$
    - ...
2. Rekurzivno rešim vse te podprobleme.
3. Rešitev problema: vzamem maksimum rešitev podproblemov.

$$2 + 4 * (-8) + 9 * 2 + (-5) * 7 + 9$$

1. Če izberem neko operacijo (na voljo imam $(n-1)!$ izbir) in jo izračunam, dobim krajši izraz istega tipa:
   - $8 * (-8) + 9 * 2 + (-5) * 7 + 9$
   - $2 + (-32) + 9 * 2 + (-5) * 7 + 9$
   - $2 + 4 * (-1) * 2 + (-5) * 7 + 9$
   - ...
2. Rekurzivno rešim vse te podprobleme.
3. Rešitev problema: vzamem maksimum rešitev podproblemov.

Časovna zahtevnost: O(n!)
[Ta funkcija narašča še hitreje kot 2^n.]

Kako naj se tega lotim? Opazimo kaj posebnega?

Kako naj se tega lotim? Opazimo kaj posebnega?

Razmišljaj dinamično!

Kako naj se tega lotim? Opazimo kaj posebnega?

Razmišljaj dinamično!

Števila so lahko tudi negativna. Predstavlja to kakšno težavo?

http://uva.onlinejudge.org

Nekaj nalog:

- #674 (Coin Change)
- #507 (Jill Rides Again)
- #714 (Copying Books)
- #10003 (Cutting Sticks)
- #481 (What Goes Up)
- #10534 (Wavio Sequence)
- #562 (Dividing Coins)
- #10911 (Forming Quiz Teams)