

Fö. 2: Giriga algoritmer.

Fö. 3: Rekursiva algoritmer.

Fö. 4: Dynamisk programmering.

Exempel 1: Stabil blandning.

Exempel 2: Exakt summa.

Exempel 3: Kortaste vägar.

Dynamisk programmering

Dynamisk programmering

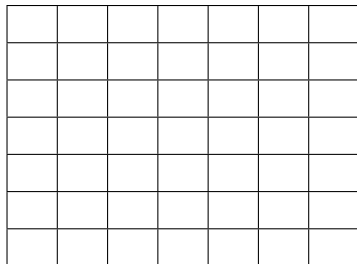
Rekursiv nedbrytning arbetar "top-down". Man försöker dela indata i ungefär lika stora delar och lösa dessa delar rekursivt.

Dynamisk programmering

Rekursiv nedbrytning arbetar "top-down". Man försöker dela indata i ungefär lika stora delar och lösa dessa delar rekursivt.

Dynamisk programmering arbetar "bottom-up". Man löser små delar av indata och kombinerar dessa successivt till en fullständig lösning.

Dynamisk programmering



Dynamisk programmering

■	■	■	■	■	■	■
■						
■						
■						
■						
■						
■						

Dynamisk programmering

■	■	■	■	■	■	■
■	■					
■						
■						
■						
■						
■						

Dynamisk programmering

■	■	■	■	■	■	■
■	■	■				
■						
■						
■						
■						
■						

Dynamisk programmering

■	■	■	■	■	■	■
■	■	■	■			
■						
■						
■						
■						
■						

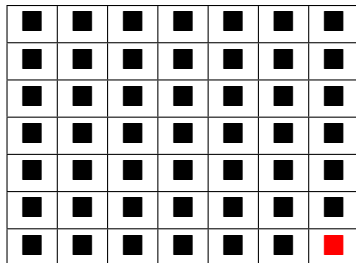
Dynamisk programmering

■	■	■	■	■	■	■
■	■	■	■	■	■	■
■						
■						
■						
■						
■						

Dynamisk programmering

■	■	■	■	■	■	■
■	■	■	■	■	■	■
■	■	■	■	■	■	■
■						
■						
■						
■						

Dynamisk programmering



Exempel 1: Stabil blandning.

Stabil blandning

Problem. Stabil blandning.

Indata. Tre teckensträngar X, Y, Z där X och Y har längd n och Z har längd $2n$.

Fråga. Är Z en stabil blandning av X och Y ?

Stabil blandning

X: abc

Y: def

Z: abdecf

Stabil blandning

X: abc

Y: def

Z: abdecf

X: abc

Y: def

Z: abdfec

Stabil blandning

X: aabb

Y: baab

Z: aabbaabb

Stabil blandning

Låt $X = X_1 \dots X_n$, $Y = Y_1 \dots Y_n$ och $Z = Z_1 \dots Z_{2n}$.

Stabil blandning

Konstruera en 0/1-tabell S .

	Y_0	Y_1	Y_2	Y_3	Y_4	$\dots$	Y_n
X_0							
X_1							
X_2							
X_3							
X_4							
$\vdots$							
X_n							

Stabil blandning

Tolkning: $S[i, j] = 1$ om och endast om $X_1 \dots X_i$ och $Y_1 \dots Y_j$ är en stabil blandning av $Z_1 \dots Z_{i+j}$.

Stabil blandning

Tolkning: $S[i, j] = 1$ om och endast om $X_1 \dots X_i$ och $Y_1 \dots Y_j$ är en stabil blandning av $Z_1 \dots Z_{i+j}$.

Rätt svar kan hittas i sista rutan $S[n, n]$.

Stabil blandning

Enkla grundfall.

$$S[0,0] = 1.$$

$$S[i+1,0] = 1 \text{ om } X_{i+1} = Z_{i+1} \text{ och } S[i,0] = 1.$$

$$S[0,j+1] = 1 \text{ om } Y_{j+1} = Z_{j+1} \text{ och } S[0,j] = 1.$$

Stabil blandning

Observation.

$$X_n = Z_{2n} \text{ eller } Y_n = Z_{2n}.$$

Stabil blandning

Observation.

$$X_n = Z_{2n} \text{ eller } Y_n = Z_{2n}.$$

Tänk på samma sätt.

$$S[i + 1, j + 1] = 1 \text{ om}$$

Stabil blandning

Observation.

$$X_n = Z_{2n} \text{ eller } Y_n = Z_{2n}.$$

Tänk på samma sätt.

$$S[i+1, j+1] = 1 \text{ om}$$

$$X_{i+1} = Z_{i+j+2} \text{ och } S(i, j+1) = 1$$

eller

$$Y_{j+1} = Z_{i+j+2} \text{ och } S(i+1, j) = 1$$

Stabil blandning

Med dessa fyra regler kan tabellen fyllas i systematiskt.

Stabil blandning

Med dessa fyra regler kan tabellen fyllas i systematiskt.

	Y_0	Y_1	Y_2	Y_3	Y_4	$\dots$	Y_n
X_0							
X_1					•		
X_2				•	■		
X_3							
X_4							
$\vdots$							
X_n							

Stabil blandning

Tidskomplexitet: $O(n^2)$.

Stabil blandning

Korrekthetsbevis (skiss).

Vi säger att tabellen är (i, j) -korrekt om alla koordinater (a, b) där $1 \leq a \leq i$ och $1 \leq b \leq j$ är korrekt ifyllda.

Vi visar att tabellen är (i, j) -korrekt för $0 \leq i, j \leq n$. Detta kan vi göra genom att visa att tabellen är (i, j) -korrekt för alla $m = i + j$, $0 \leq i, j \leq n$.

Stabil blandning

Induktion över $m = i + j$.

Bassteg. Om $m = 0$ så är tabellen $(0,0)$ -korrekt och algoritmen ger rätt svar.

Ind. ant. Antag att tabellen är (i,j) -korrekt för alla $i + j \leq m \leq k$, $k \geq 0$.

Stabil blandning

Ind. steg. Antag att $i + j = k + 1$,

Kom ihåg att

$$S[i, j] = 1 \text{ om}$$

$$X_i = Z_{i+j} \text{ och } S(i-1, j) = 1$$

eller

$$Y_j = Z_{i+j} \text{ och } S(i, j-1) = 1$$

Notera att att $i-1+j = k$ och $i+j-1 = k$. Alltså är $S(i-1, j)$ och $S(i, j-1)$ korrekt beräknade via ind. ant.

Exempel 2: Exakt summa.

Exakt summa

Problem. Exakt summa.

Indata. En mängd I med positiva heltal sådan att $1 \in I$. Ett positivt heltal x .

Utdata. En kortast möjliga sekvens $S = (i_1, \dots, i_k)$ av tal från I sådan att $\sum_{j=1}^k s_j = x$.

Exakt summa

Observation 1: Vi behöver bara använda tal $i \in I$ som uppfyller $i \leq x$.

Exakt summa

Observation 1: Vi behöver bara använda tal $i \in I$ som uppfyller $i \leq x$.

Observation 2: Sekvensens längd är maximalt x eftersom $1 \in I$.

Exakt summa

Konstruera en 0/1-tabell R .

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$\dots$	$j = x$
$i = 1$							
$i = 2$							
$i = 3$							
$i = 4$							
$i = 5$							
$\vdots$							
$i = x$							

Exakt summa

Tolkning.

$R(i, j) = 1$ om talet i kan konstrueras med j tal ur I .

Exakt summa

Enkelt grundfall.

$$R(i, 1) = 1 \text{ omm } i \in I.$$

Exakt summa

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$\dots$	$j = x$
$i = 1$	■						
$i = 2$	■						
$i = 3$	■						
$i = 4$	■						
$i = 5$	■						
$\dots$	■						
$i = x$	■						

Exakt summa

Uppdatering.

$R(i, j) = 1$ omm det existerar $t \in I$ sådant att det finns ett $1 \leq i' \leq x$ sådant att $R(i', j - 1) = 1$ och $i' + t = i$.

Exakt summa

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$\dots$	$j = x$
$i = 1$	■	■					
$i = 2$	■	■					
$i = 3$	■	■					
$i = 4$	■	■					
$i = 5$	■	■					
$\dots$	■	■					
$i = x$	■	■					

Exakt summa

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$\dots$	$j = x$
$i = 1$	■	■	■				
$i = 2$	■	■	■				
$i = 3$	■	■	■				
$i = 4$	■	■	■				
$i = 5$	■	■	■				
$\dots$	■	■	■				
$i = x$	■	■	■				

Exakt summa

Den första 1:an på rad x indikerar den optimala lösningen.

Exakt summa

Den första 1:an på rad x indikerar den optimala lösningen.

Hur får man fram konkret lösning?

Exakt summa

Den första 1:an på rad x indikerar den optimala lösningen.

Hur får man fram konkret lösning?

Tidskomplexitet: $O(\text{poly}(x, |I|))$.

Exempel 3: Kortaste vägar.

Kortaste vägar

	v_1	v_2	v_3	v_4
v_1	0	?	3	?
v_2	2	0	?	?
v_3	?	6	0	5
v_4	?	2	?	0

Kortaste vägar

	v_1	v_2	v_3	v_4
v_1	0	?	3	?
v_2	2	0	?	?
v_3	?	6	0	5
v_4	?	2	?	0

	v_1	v_2	v_3	v_4
v_1	0	9	3	8
v_2	2	0	5	10
v_3	8	6	0	5
v_4	4	2	7	0

Kortaste vägar

D är $n \times n$ -matris över $\mathbb{Q}^+ \cup \{\infty\}$.

Förenklar att ersätta "?" med " ∞ ".

Kortaste vägar

for $k = 1$ to n do

Kortaste vägar

```
for  $k = 1$  to  $n$  do  
  for  $i = 1$  to  $n$  do  
    for  $j = 1$  to  $n$  do
```

Kortaste vägar

```
for  $k = 1$  to  $n$  do  
  for  $i = 1$  to  $n$  do  
    for  $j = 1$  to  $n$  do  
       $D(i, j) = \min(D(i, j), D(i, k) + D(k, j))$ 
```

Kortaste vägar

```
for  $k = 1$  to  $n$  do
  for  $i = 1$  to  $n$  do
    for  $j = 1$  to  $n$  do
       $D(i, j) = \min(D(i, j), D(i, k) + D(k, j))$ 
return  $D$ 
```

Kortaste vägar

Tidskomplexitet: $O(n^3)$.

Kortaste vägar

Korrekthetsbevis.

En k -väg från v_a till v_b är en väg

$$v_a \rightarrow v_{i_1} \rightarrow \cdots \rightarrow v_{i_m} \rightarrow v_b$$

sådan att

$$\{v_{i_1}, \dots, v_{i_m}\} \subseteq \{v_1, \dots, v_k\}.$$

Kortaste vägar

Korrekthetsbevis.

En k -väg från v_a till v_b är en väg

$$v_a \rightarrow v_{i_1} \rightarrow \cdots \rightarrow v_{i_m} \rightarrow v_b$$

sådan att

$$\{v_{i_1}, \dots, v_{i_m}\} \subseteq \{v_1, \dots, v_k\}.$$

Notera att en n -väg från v_a till v_b är vilken väg som helst från v_a till v_b .

Kortaste vägar

Låt $D^{(k)}[i,j]$ vara längden av den kortaste k -vägen från v_i till v_j .

Kortaste vägar

Låt $D^{(k)}[i,j]$ vara längden av den kortaste k -vägen från v_i till v_j .

Vi visar att algoritmen beräknar $D^{(1)}, D^{(2)}, \dots, D^{(n)}$ och alltså returnerar $D^{(n)}$ (vilket är korrekt svar).

Kortaste vägar

Induktion över antalet varv m som yttersta **for**-slingan har gjort.

Kortaste vägar

Induktion över antalet varv m som yttersta **for**-slingan har gjort.

Bassteg. $m = 0$. $D^{(0)} = D$, dvs direktvägarna mellan noderna.

Kortaste vägar

Induktion över antalet varv m som yttersta **for**-slingan har gjort.

Bassteg. $m = 0$. $D^{(0)} = D$, dvs direktvägarna mellan noderna.

Ind. ant. Antag att $m = k - 1$, $k - 1 \geq 0$, och att $D^{(k-1)}$ har blivit korrekt beräknad.

Kortaste vägar

Induktion över antalet varv m som yttersta **for**-slingan har gjort.

Bassteg. $m = 0$. $D^{(0)} = D$, dvs direktvägarna mellan noderna.

Ind. ant. Antag att $m = k - 1$, $k - 1 \geq 0$, och att $D^{(k-1)}$ har blivit korrekt beräknad.

Ind. steg. Algoritmen beräknar

$$D^{(k)}(i, j) = \min(D^{(k-1)}(i, j), D^{(k-1)}(i, k) + D^{(k-1)}(k, j))$$

Detta är längden av kortaste k -vägen mellan v_i och v_j . Antingen används v_k inte på denna väg (första fallet) eller så gör den det (andra fallet). Notera att det inte finns anledning att använda v_k flera gånger på den kortaste k -vägen.