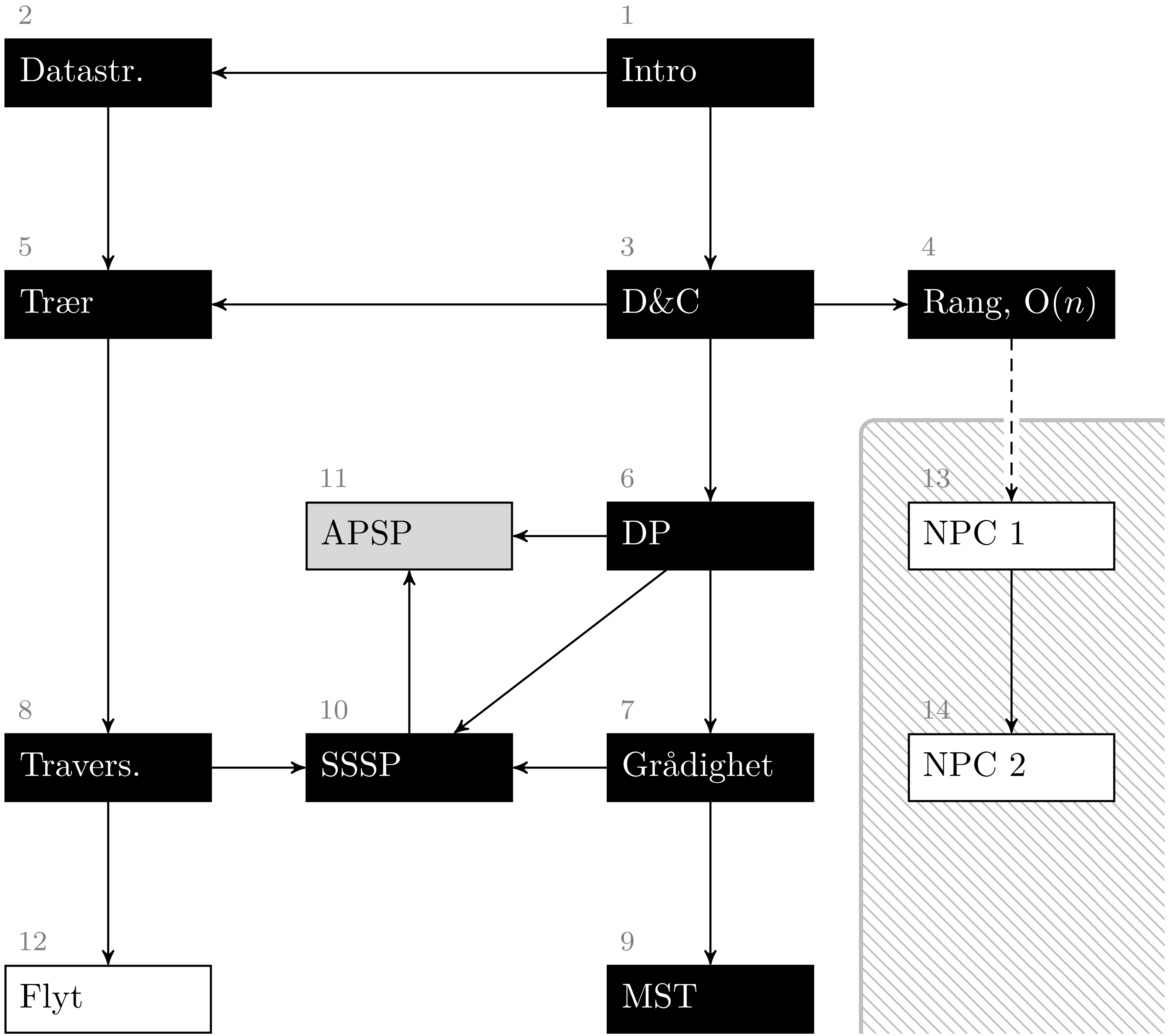
Forelesning 11

Korteste vei fra alle til alle

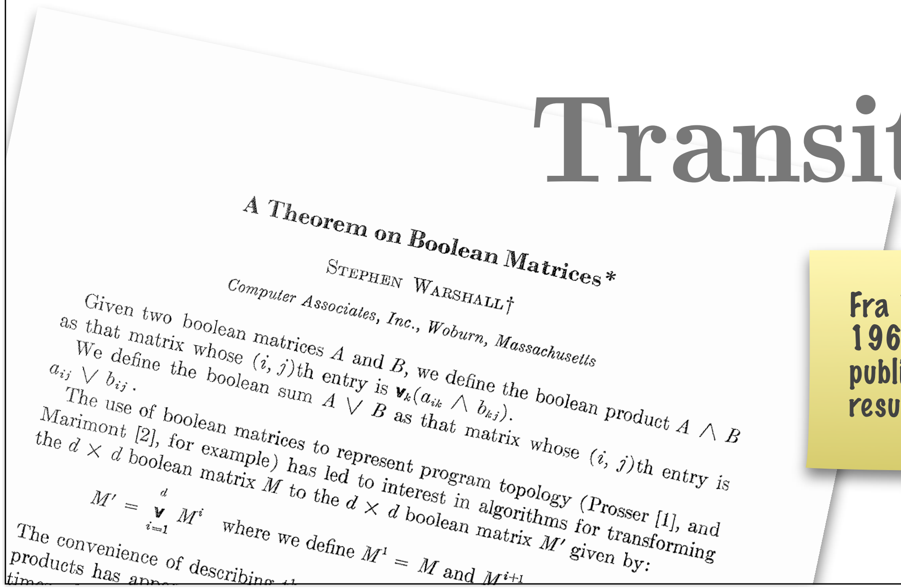

2
Datastr.
1
Intro
5
Trær
3
D&C
4
Rang, O(n)
11
APSP
6
DP
13
NPC 1
8
Travers.
10
SSSP
7
Grådighet
14
NPC 2
12
Flyt
9
MST

1. **Transitiv lukning**

2. **Floyd-Warshall**

1:2

Transitiv lukning



A Theorem on Boolean Matrices*

Stephen Warshall†

Computer Associates, Inc., Woburn, Massachusetts

Given two boolean matrices A and B, we define the boolean product $A \wedge B$ as that matrix whose (i, j)th entry is $\vee_k (a_{ik} \wedge b_{kj})$.

We define the boolean sum $A \vee B$ as that matrix whose (i, j)th entry is $a_{ij} \vee b_{ij}$.

The use of boolean matrices to represent program topology (Prosser [1], and Marimont [2], for example) has led to interest in algorithms for transforming the $d \times d$ boolean matrix M to the $d \times d$ boolean matrix M' given by:

$M' = \bigvee_{i=1}^{d} M^i$ where we define $M^1 = M$ and M^{i+1}

The convenience of describing ...

Fra 1960 (publisert 1962). Bernhard Roy publiserte samme resultat separat i 1959.

Input: En rettet graf $G = (V, E)$.

Output: En rettet graf $G^* = (V, E^*)$ der $(i, j) \in E^*$ hvis og bare hvis det finnes en sti fra i til j i G.

Vi kommer til å returnere en nabomatrise T for G^*.

Traversér fra hver node?

- Kjøretid: $\mathrm{V} \times \Theta(\mathrm{E} + \mathrm{V}) = \Theta(\mathrm{VE} + \mathrm{V}^2)$
- Bra når vi har få kanter, f.eks. $\mathrm{E} = o(\mathrm{V}^2)$
- Mye overhead; høye konstantledd

Målsetting:

- Vi fokuserer på tilfellet $\mathrm{E} = \Theta(\mathrm{V}^2)$
- Vi vil ha et lavere konstantledd

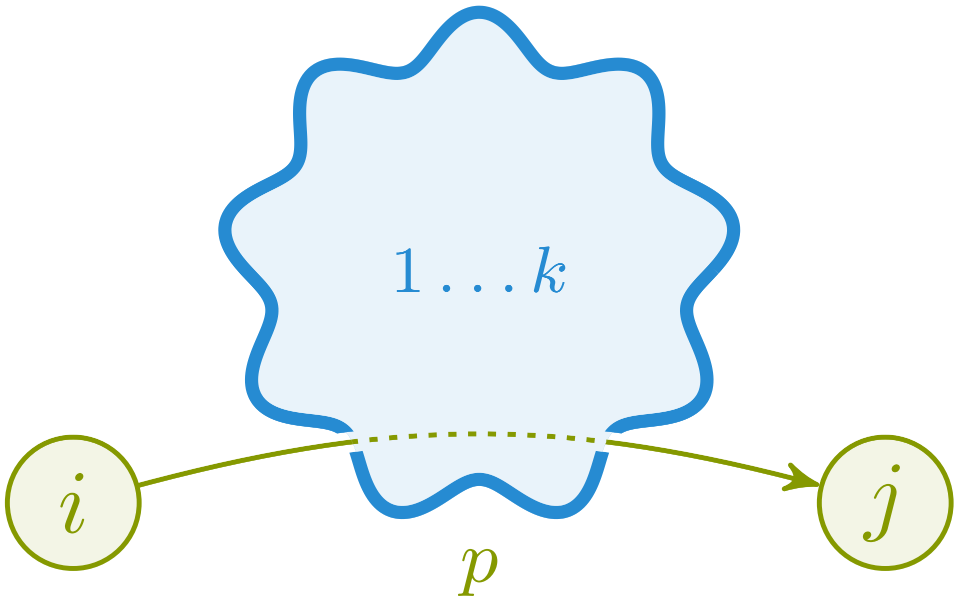
Observasjon:

- Korteste stier har felles segmenter
- Overlappende delproblemer . . .

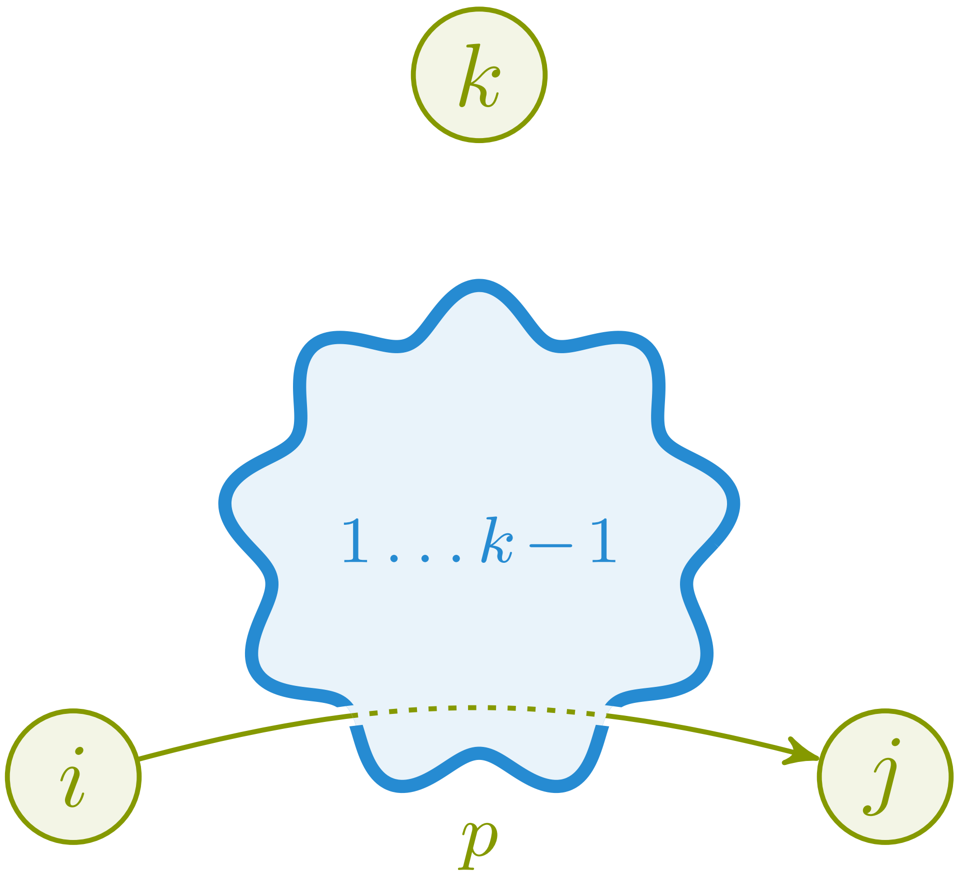
Dekomponering: Hva blir «koordinatene» til delproblemene?

$$t_{ij}^{(k)}$$

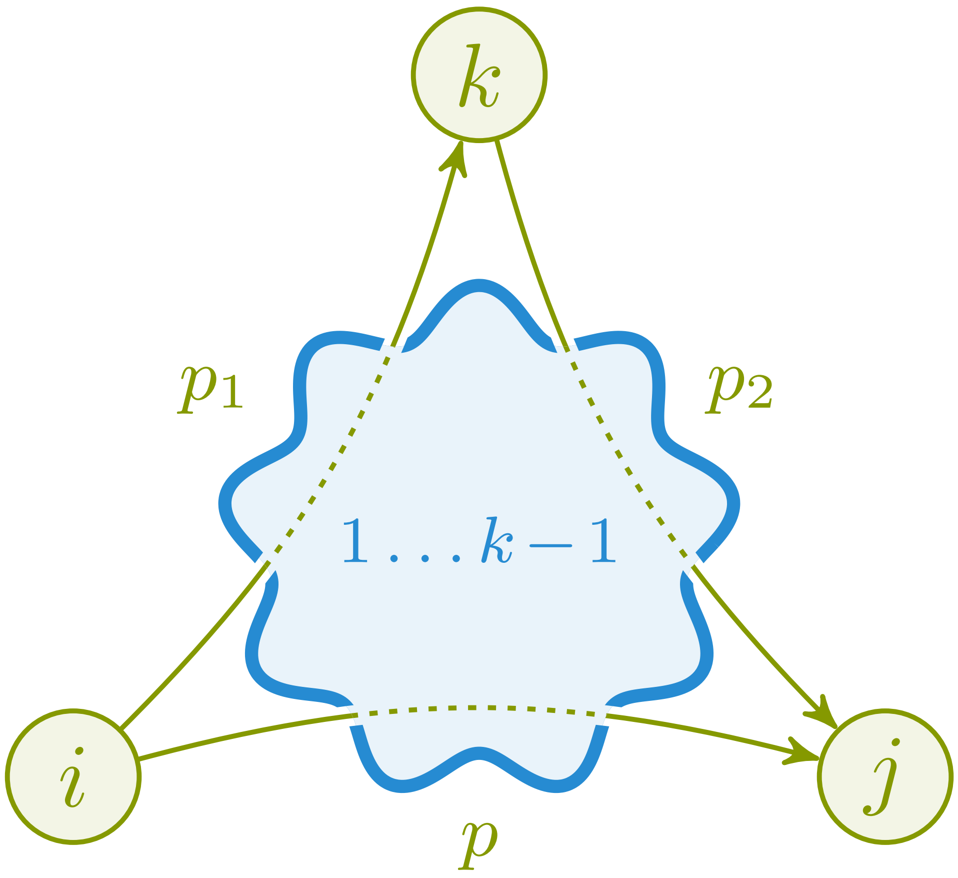
Det finnes en vei fra i til j via noder fra $\{1 \ldots k\}$



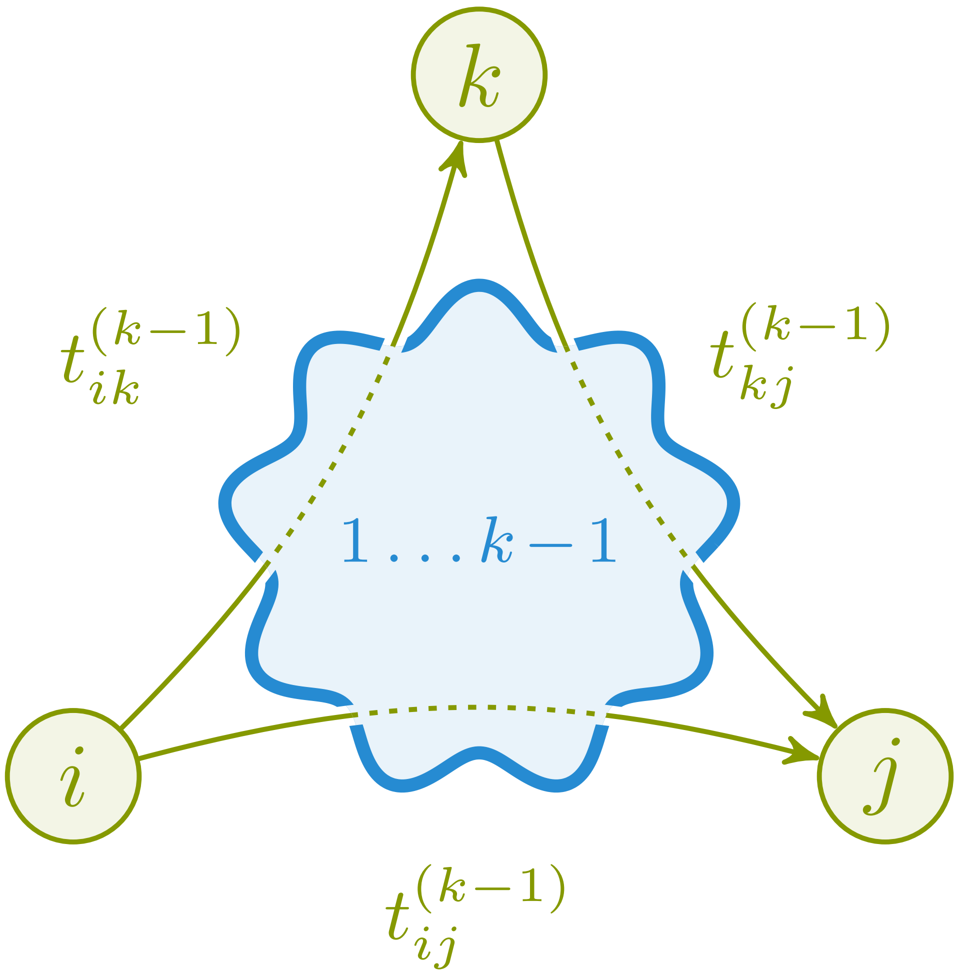
$t_{ij}^{(k)}$ = det går en vei fra i til j via noder fra $\{1 \ldots k\}$



Som for ryggsekkproblemet: Skal k være med eller ikke?



De mulige stiene p, p_1 og p_2 går kun via noder fra $\{1 \ldots k-1\}$



$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}\right)$$

$$t_{ij}^{(0)} = \begin{cases} 0 & \text{if } i \neq j \text{ and } (i,j) \notin \mathrm{E}\,, \\ 1 & \text{if } i = j \text{ or } (i,j) \in \mathrm{E}\,. \end{cases}$$

$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}\right)$$

$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$$

Problemet er at iterasjonene 1...k «blandes», så vi kan risikere at noen av del-stiene allerede går innom k – så kanskje vi går innom k mer enn én gang? I så fall har vi en sykel ... men om det finnes en sti *med* en sykel, så finnes det også en sti *uten* en sykel!

Det er trygt å bruke én tabell. Hvorfor?

Transitive-Closure(G)

For hver node i og j: Finnes det en sti $i \leadsto j$?

Transitive-Closure(G)

1 $n = |G.V|$
2 let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix

Finnes en sti $i \rightsquigarrow j$ som ikke går via andre noder?

TRANSITIVE-CLOSURE(G)

$n = |\text{G.V}|$

let $\text{T}^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix

for $i = 1$ **to** n

For hver mulig startnode . . .

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
  for $j = 1$ **to** n

For hver mulig sluttnode…

```
TRANSITIVE-CLOSURE(G)
n = |G.V|
let T^(0) = (t_ij^(0)) be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        if i == j or (i, j) ∈ G.E
```

Samme node eller sammenkoblet med kant?

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$

Da finnes det en sti $i \rightsquigarrow j$ som ikke går via andre noder

```
TRANSITIVE-CLOSURE(G)
n = |G.V|
let T^(0) = (t_ij^(0)) be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        if i == j or (i, j) ∈ G.E
            t_ij^(0) = 1
        else t_ij^(0) = 0
```

Ellers finnes ingen slik sti

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n

Vi får nå lov til å gå innom node k

Transitive-Closure(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n
    let $T^{(k)} = (t_{ij}^{(k)})$ be a new $n \times n$ matrix

Finnes en sti $i \leadsto j$ som kun får gå innom $\{1, \dots, k\}$?

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n
    let $T^{(k)} = (t_{ij}^{(k)})$ be a new $n \times n$ matrix
    for $i = 1$ **to** n

For hver mulig startnode . . .

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n
    let $T^{(k)} = (t_{ij}^{(k)})$ be a new $n \times n$ matrix
    for $i = 1$ **to** n
        for $j = 1$ **to** n

For hver mulig sluttnode . . .

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n
    let $T^{(k)} = (t_{ij}^{(k)})$ be a new $n \times n$ matrix
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$

Finnes $i \rightsquigarrow j$ eller $i \rightsquigarrow k \rightsquigarrow j$, om vi kun får gå innom $\{1, \ldots, k-1\}$?

TRANSITIVE-CLOSURE(G)

$n = |G.V|$
let $T^{(0)} = (t_{ij}^{(0)})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        if $i == j$ or $(i, j) \in G.E$
            $t_{ij}^{(0)} = 1$
        else $t_{ij}^{(0)} = 0$
for $k = 1$ **to** n
    let $T^{(k)} = (t_{ij}^{(k)})$ be a new $n \times n$ matrix
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$
return $T^{(n)}$

Finnes det en sti $i \rightsquigarrow j$ som får gå innom $\{1, \ldots, n\}$, dvs. alle?

$\textsc{Transitive-Closure}'(G)$

Forenklet utgave: Bruker bare én tabell, T

```
TRANSITIVE-CLOSURE′(G)
  n = |G.V|
  initialize T
```

Samme node (I er identitetsmatrise) eller nabo (A er nabomatrise)

TRANSITIVE-CLOSURE′(G)
1 $n = |G.V|$
2 initialize T
3 **for** $k = 1$ **to** n

Vi får nå lov til å gå innom node k

```
Transitive-Closure′(G)
n = |G.V|
initialize T
for k = 1 to n
        for i = 1 to n
```

For hver mulig startnode . . .

```
Transitive-Closure′(G)
n = |G.V|
initialize T
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
```

For hver mulig sluttnode . . .

```
TRANSITIVE-CLOSURE′(G)
n = |G.V|
initialize T
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            t_ij = t_ij ∨ (t_ik ∧ t_kj)
```

Enten allerede en sti $i \rightsquigarrow j$ eller en ny sti $i \rightsquigarrow k \rightsquigarrow j$?

```
TRANSITIVE-CLOSURE′(G)
n = |G.V|
initialize T
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            t_ij = t_ij ∨ (t_ik ∧ t_kj)
return T
```

Finnes det en sti $i \rightsquigarrow j$?

```
Transitive-Closure′(G)
n = |G.V|
initialize T
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            t_ij = t_ij ∨ (t_ik ∧ t_kj)
return T
```

$k, i, j \; = \; 1, 1, 1$

	1	2	3	4	5
1	1		1		
2	1	1		1	
3			1		1
4				1	
5		1	1	1	1

```
TRANSITIVE-CLOSURE′(G)
n = |G.V|
initialize T
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            t_ij = t_ij ∨ (t_ik ∧ t_kj)
return T
```

	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4				1	
5	1	1	1	1	1

$k, i, j \;=\; 5, 5, 5$

Korteste vei ›

Transitiv lukning › **Kjøretid**

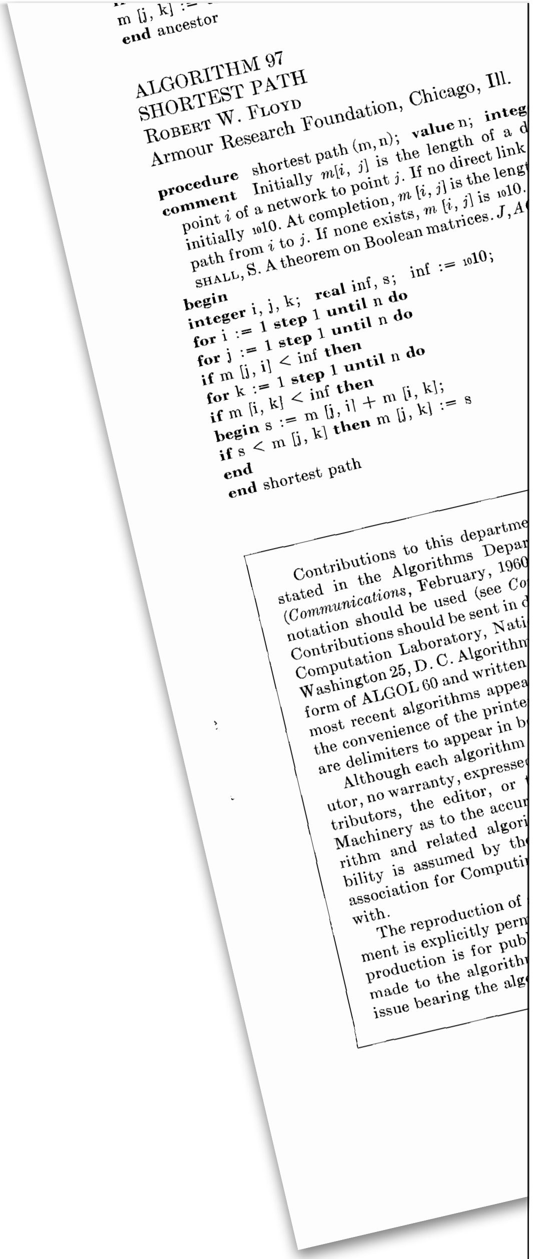
init › $\Theta(n^2)$
for $i = 1$ **to** n
  for $j = 1$ **to** n
    sett $t_{ij}^{(0)}$ › $O(1)$
for $k = 1$ **to** n
  evt. ny matrise › $\Theta(n^2)$
  for $i = 1$ **to** n
    for $j = 1$ **to** n
      sett $t_{ij}^{(k)}$ › $O(1)$
return › $O(1)$

Totalt: $\Theta(n^3)$

Fra 1962.

2:2

Floyd-Warshall


ALGORITHM 97
SHORTEST PATH
ROBERT W. FLOYD
Armour Research Foundation, Chicago, Ill.

```
procedure shortest path (m,n); value n; integ
comment Initially m[i, j] is the length of a d
  point i of a network to point j. If no direct link
  initially 10^10. At completion, m [i, j] is the leng
  path from i to j. If none exists, m [i, j] is 10^10.
  SHALL, S. A theorem on Boolean matrices. J, A
begin
integer i, j, k; real inf, s; inf := 10^10;
for i := 1 step 1 until n do
for j := 1 step 1 until n do
if m [j, i] < inf then
for k := 1 step 1 until n do
if m [i, k] < inf then
begin s := m [j, i] + m [i, k];
if s < m [j, k] then m [j, k] := s
end
end shortest path
```

Contributions to this departme stated in the Algorithms Depar (Communications, February, 1960 notation should be used (see Co Contributions should be sent in d Computation Laboratory, Nati Washington 25, D. C. Algorithm form of ALGOL 60 and written most recent algorithms appea the convenience of the printe are delimiters to appear in b Although each algorithm utor, no warranty, expresse tributors, the editor, or Machinery as to the accur rithm and related algori bility is assumed by the association for Computi with.
The reproduction of ment is explicitly perm production is for pub made to the algorith issue bearing the alg

Input: En vektet, rettet graf $G = (V, E)$ uten negative sykler, der $V = \{1, \dots, n\}$, og vektene er gitt av matrisen $W = (w_{ij})$.

Output: En $n \times n$-matrise $D = (d_{ij})$ med avstander, dvs., $d_{ij} = \delta(i, j)$.

Vi returnerer også en forgjengermatrise $\Pi = (\pi_{ij})$

Fra hver node til alle andre?

- Dijkstra med tabell: $O(V^3 + VE)$
- ... med binærhaug: $O(VE \lg V)$
- ... med Fib.-haug: $O(V^2 \lg V + VE)$
- Bellman-Ford: $\Theta(V^2 E)$

Målsetting:

- Vi vil tillate negative kanter
- Vi vil ha lavere asymptotisk kjøretid ...
- ... og vil ha lavere konstantledd

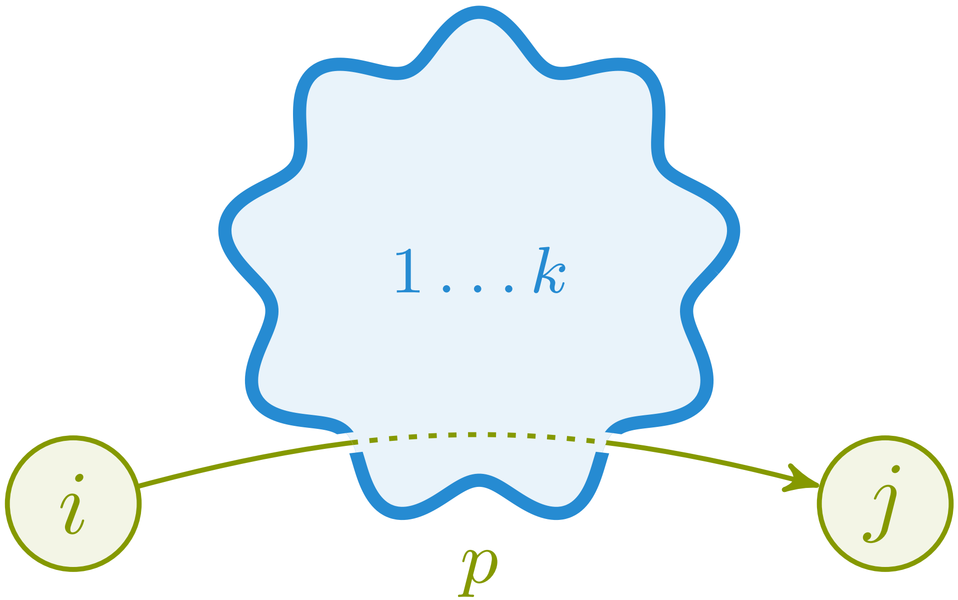
Nok en gang: Hva blir «koordinatene» til delproblemene?

$$d_{ij}^{(k)}$$

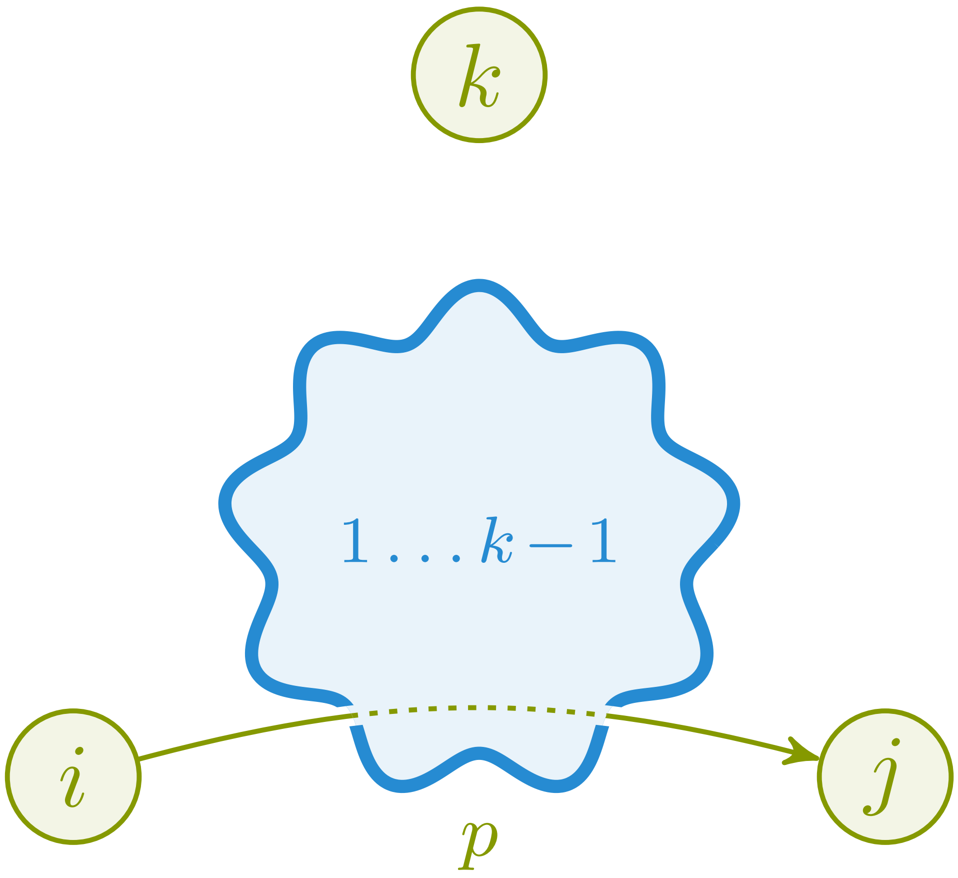
Korteste vei fra i til j via noder fra $\{1 \ldots k\}$

$$\pi_{ij}^{(k)}$$

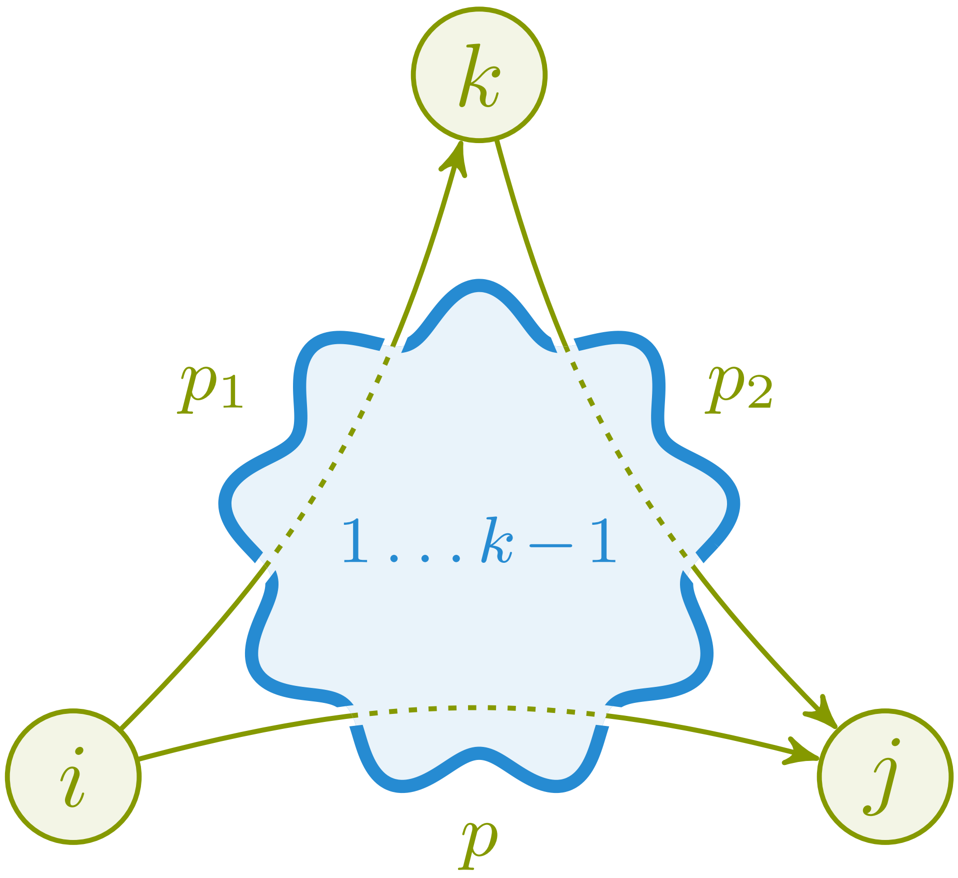
Forgjengeren til j om vi starter i i og går via noder fra $\{1 \ldots k\}$



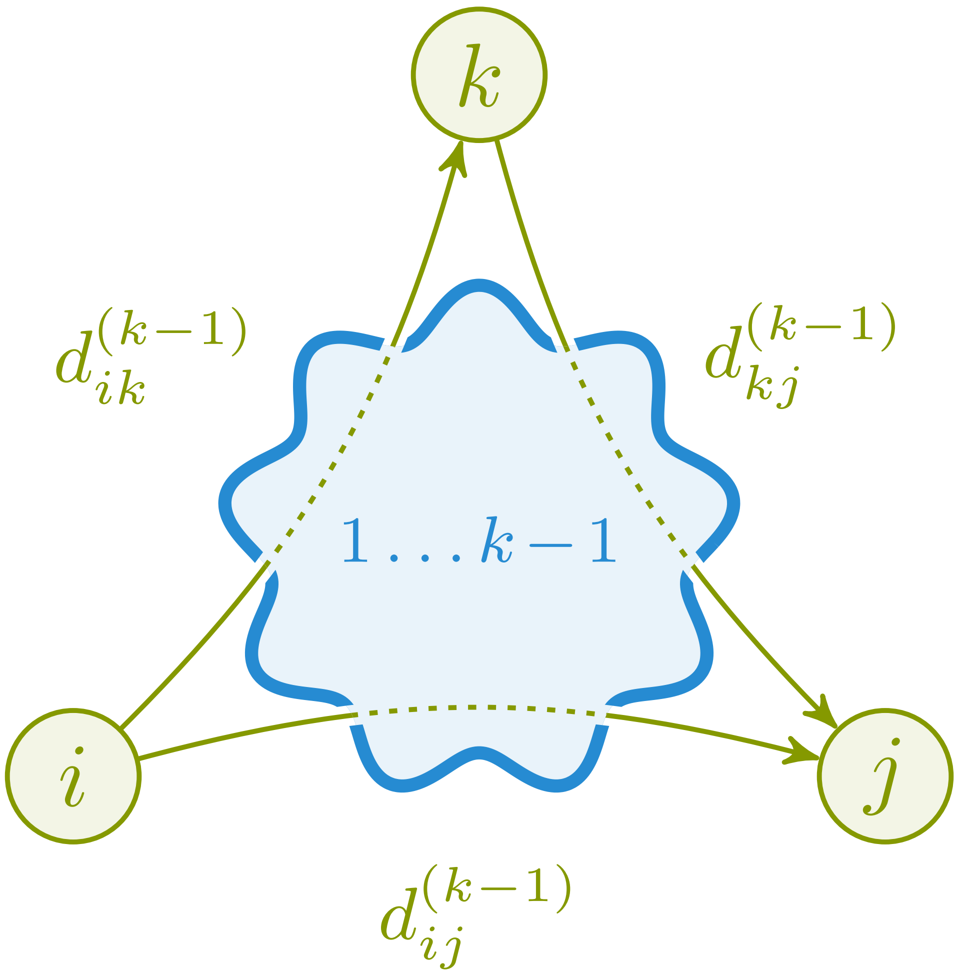
$d_{ij}^{(k)}$ = korteste vei fra i til j via noder fra $\{1 \dots k\}$



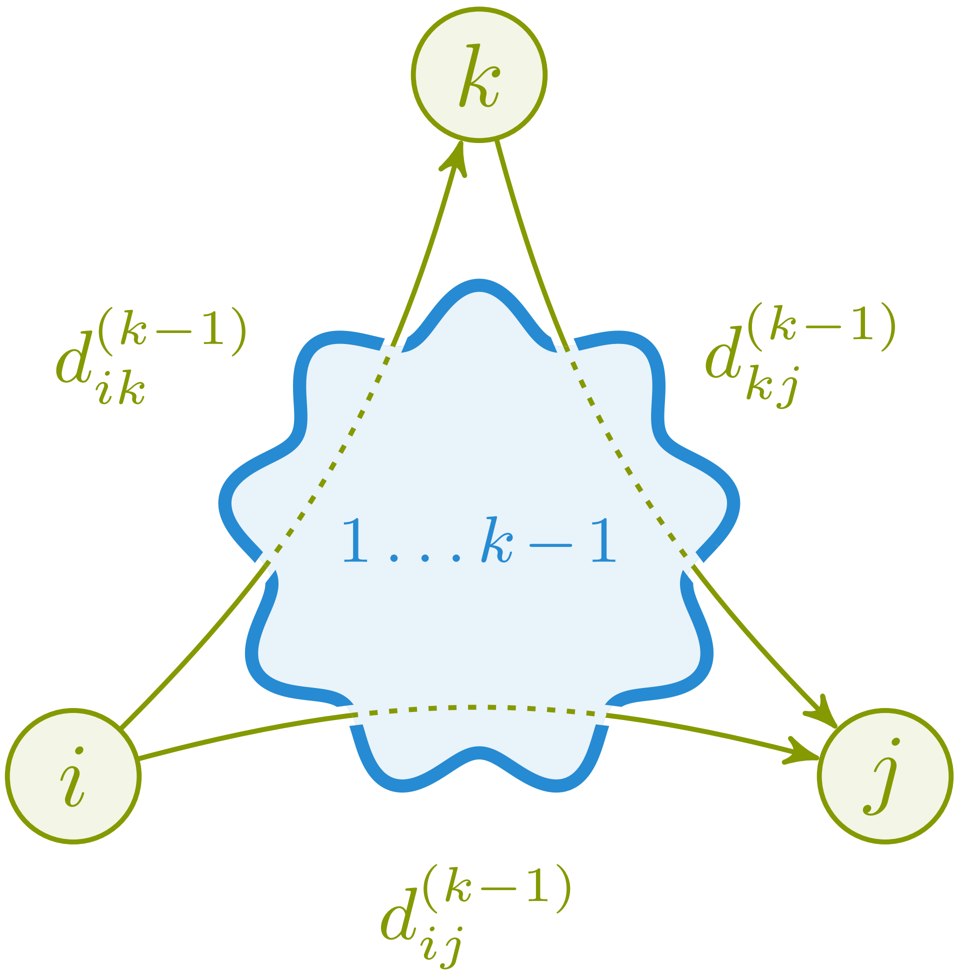
Som for ryggsekkproblemet: Skal k være med eller ikke?



Stiene p, p_1 og p_2 går kun via noder fra $\{1 \ldots k-1\}$



$d_{ij}^{(k)}$ kan enten være $d_{ij}^{(k-1)}$ eller $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$



$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)},\, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty\,, \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty\,. \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0\,, \\ \min\left(d_{ij}^{(k-1)},\, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right) & \text{if } k \geq 1\,. \end{cases}$$

Som før: Vi kan ha gått innom k i én av delstiene, siden vi blander iterasjoner – men vi antar at det ikke er noen negative sykler, og en positiv sykel vil aldri lønne seg (og vil dermed ikke bli med).

Vi kan bruke én tabell igjen (se oppgave 25.2-4)

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,, \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\,. \end{cases}$$

Fordi denne bare baserer seg
på valget vi gjør for d.

Også her holder det med én tabell

FLOYD-WARSHALL(W)

For hver node i og j: Hva er korteste vei $i \rightsquigarrow j$?

FLOYD-WARSHALL(W)

$n = W.rows$

$D^{(0)} = W$

Korteste vei $i \rightsquigarrow j$ som ikke går via andre $= w(i, j)$

Floyd-Warshall(W)

$n = W.\mathit{rows}$
$D^{(0)} = W$
for $k = 1$ **to** n

Vi får nå lov til å gå innom node k

FLOYD-WARSHALL(W)

$n = \mathrm{W}.rows$

$\mathrm{D}^{(0)} = \mathrm{W}$

for $k = 1$ **to** n

  let $\mathrm{D}^{(k)} = (d_{ij}^{(k)})$ be a new $n \times n$ matrix

Hva er korteste vei $i \rightsquigarrow j$ som kun får gå innom $\{1, \ldots, k\}$?

FLOYD-WARSHALL(W)
$n = W.rows$
$D^{(0)} = W$
for $k = 1$ **to** n
    let $D^{(k)} = (d_{ij}^{(k)})$ be a new $n \times n$ matrix
    for $i = 1$ **to** n

For hver mulig startnode . . .

```
FLOYD-WARSHALL(W)
n = W.rows
D^(0) = W
for k = 1 to n
    let D^(k) = (d_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
```

For hver mulig sluttnode . . .

```
FLOYD-WARSHALL(W)
n = W.rows
D^(0) = W
for k = 1 to n
    let D^(k) = (d_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
            d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
```

Er det raskere å gå innom k (og ellers fortsatt kun $\{1, \ldots, k-1\}$)?

Floyd-Warshall(W)

$n = \mathrm{W}.rows$

$\mathrm{D}^{(0)} = \mathrm{W}$

for $k = 1$ **to** n

  let $\mathrm{D}^{(k)} = (d_{ij}^{(k)})$ be a new $n \times n$ matrix

  for $i = 1$ **to** n

    for $j = 1$ **to** n

      $d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$

return $\mathrm{D}^{(n)}$

Korteste vei $i \rightsquigarrow j$ som får gå innom $\{1, \ldots, n\}$, dvs. alle

FLOYD-WARSHALL′(W)

Forenklet: Bruker bare D heller enn $D^{(k)}$ (se 25.2-4, uten Π)

FLOYD-WARSHALL′(W)

```
1  n = W.rows
2  initialize D and Π
```

Ikke via noen: $d_{ij} = w(i, j)$; $\pi_{ij} = i$ når $i \neq j$ og $w(i, j) < \infty$

FLOYD-WARSHALL$'$(W)
1 $n = \mathrm{W}.rows$
2 initialize D and Π
3 **for** $k = 1$ **to** n

Vi får nå lov til å gå innom node k

```
FLOYD-WARSHALL′(W)
n = W.rows
initialize D and Π
for k = 1 to n
        for i = 1 to n
```

For hver mulig startnode . . .

```
FLOYD-WARSHALL′(W)
n = W.rows
initialize D and Π
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
```

For hver mulig sluttnode…

```
FLOYD-WARSHALL′(W)
n = W.rows
initialize D and Π
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            if d_ij > d_ik + d_kj
```

Er det raskere å gå innom k (og ellers fortsatt kun $\{1, \ldots, k-1\}$)?

Floyd-Warshall′(W)

$n = W.rows$
initialize D and Π
for $k = 1$ **to** n
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            if $d_{ij} > d_{ik} + d_{kj}$
                $d_{ij} = d_{ik} + d_{kj}$

Oppdatér avstanden

```
Floyd-Warshall′(W)
n = W.rows
initialize D and Π
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            if d_ij > d_ik + d_kj
                d_ij = d_ik + d_kj
                π_ij = π_kj
```

Husk valget: Forgengeren til i om vi kommer fra j

```
FLOYD-WARSHALL′(W)
n = W.rows
initialize D and Π
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            if d_ij > d_ik + d_kj
                d_ij = d_ik + d_kj
                π_ij = π_kj
return D, Π
```

$k, i, j \;=\; 1, 1, 1$

D

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

Π

	1	2	3	4	5
1		1	1		1
2				2	2
3		3			
4	4		4		
5				5	

Floyd-Warshall′(W)

$n = \text{W}.rows$
initialize D and Π
for $k = 1$ **to** n
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            if $d_{ij} > d_{ik} + d_{kj}$
                $d_{ij} = d_{ik} + d_{kj}$
                $\pi_{ij} = \pi_{kj}$
return D, Π

$k, i, j \;=\; 5, 5, 5$

D

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

Π

	1	2	3	4	5
1		3	4	5	1
2	4		4	2	1
3	4	3		2	1
4	4	3	4		1
5	4	3	4	5	

PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)

Hvilke noder er i stien fra i til j?

PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
1 **if** $i == j$

Samme node? Kommer ingenstedsfra …

```
Print-All-Pairs-Shortest-Path(Π, i, j)
1  if i == j
2      print i
```

… så bare skriv ut noden

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
if i == j
    print i
elseif π_ij == NIL
```

Hvis vi ellers ikke kom fra noe sted . . .

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
if i == j
    print i
elseif π_ij == NIL
    print "no path from" i "to" j "exists"
```

… så finnes ingen sti!

PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
if $i == j$
    print i
elseif π_{ij} == NIL
    print "no path from" i "to" j "exists"
else PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, π_{ij})

Ellers kom vi fra π_{ij}. Skriv veien dit først . . .

Print-All-Pairs-Shortest-Path(Π, i, j)

if $i == j$

  print i

elseif $\pi_{ij} ==$ NIL

  print "no path from" i "to" j "exists"

else Print-All-Pairs-Shortest-Path(Π, i, π_{ij})

  print j

… og deretter sluttnoden

PRINT-PATH(Π, i, j)

if $i == j$
    print i
elseif π_{ij} == NIL
    print "no such path"
else PRINT-PATH(Π, i, π_{ij})
    print j

Π

	1	2	3	4	5
1		3	4	5	1
2	4		4	2	1
3	4	3		2	1
4	4	3	4		1
5	4	3	4	5	

$i, j \; = \; 1, 2$

```
PRINT-PATH(Π, i, j)
if i == j
    print i
elseif π_ij == NIL
    print "no such path"
else PRINT-PATH(Π, i, π_ij)
    print j
```

Π

	1	2	3	4	5
1		3	4	5	1
2	4		4	2	1
3	4	3		2	1
4	4	3	4		1
5	4	3	4	5	

$i, j = 1, 2$

1 5 4 3 2

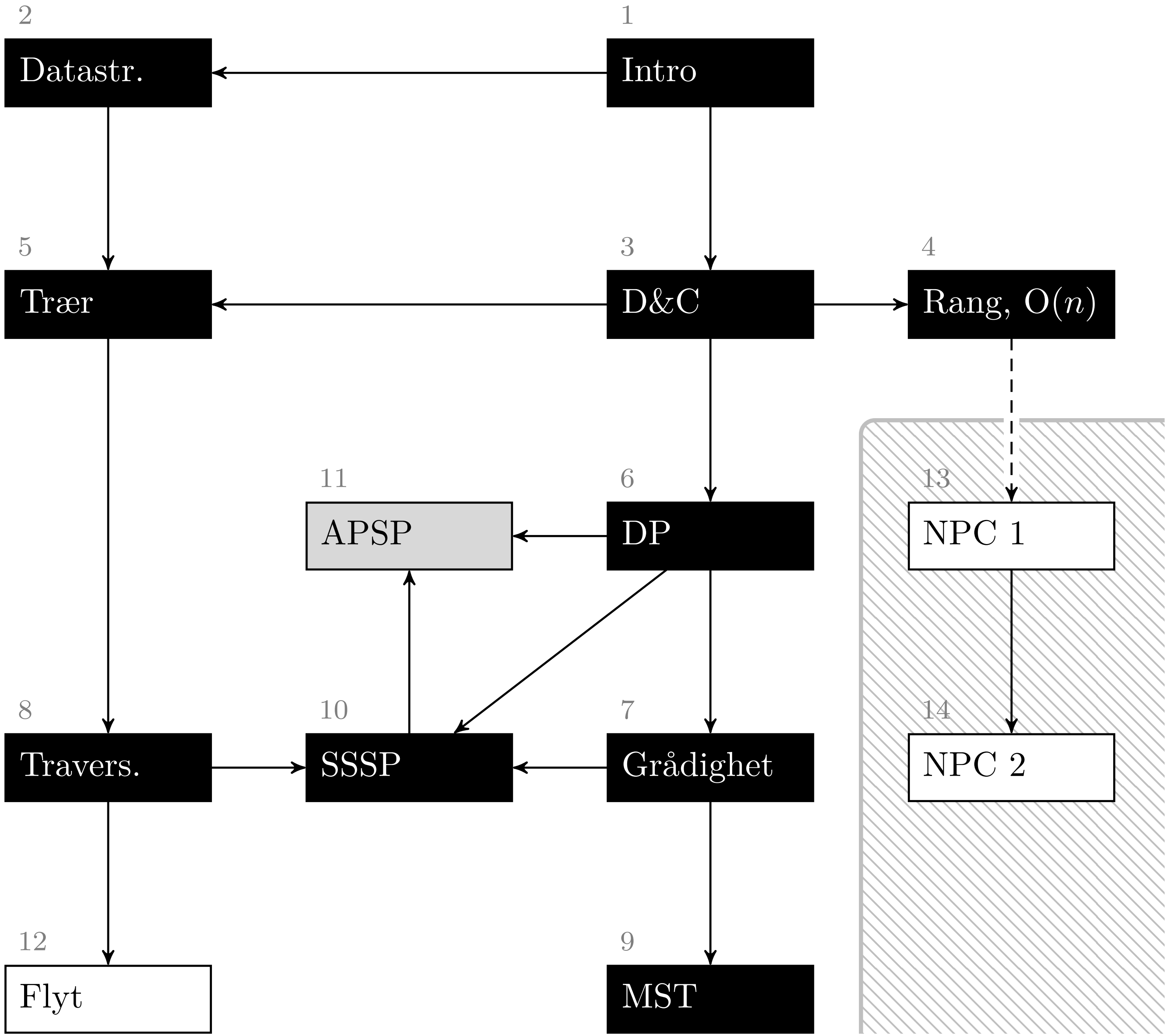
Korteste vei › Floyd-Warshall › Kjøretid

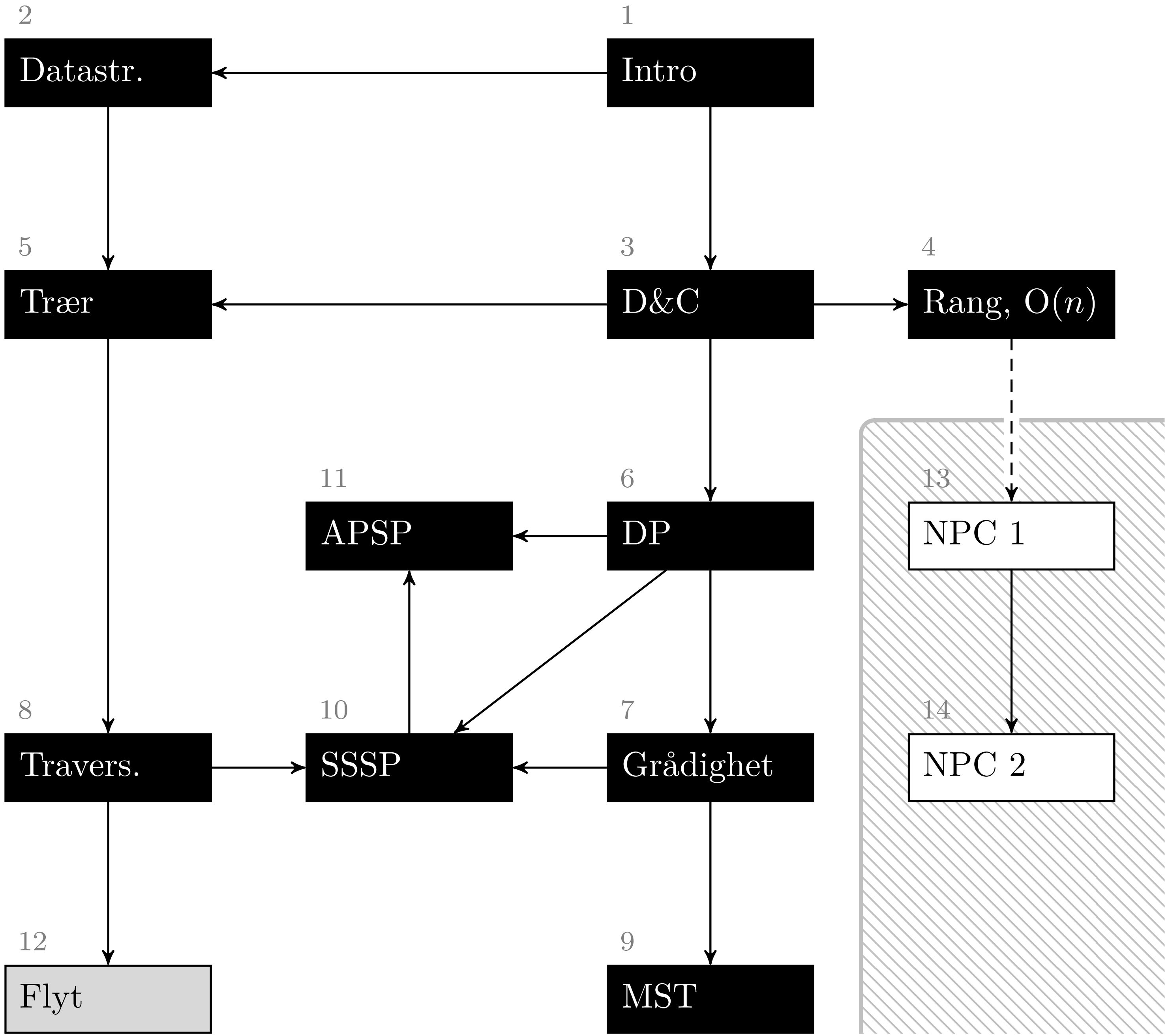
init › $\Theta(n^2)$
for $k = 1$ **to** n
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            min › $O(1)$
return › $O(1)$

Totalt: $\Theta(n^3)$

1. Transitiv lukning

2. Floyd-Warshall


2
Datastr.
1
Intro
5
Trær
3
D&C
4
Rang, O(n)
11
APSP
6
DP
13
NPC 1
8
Travers.
10
SSSP
7
Grådighet
14
NPC 2
12
Flyt
9
MST


2
Datastr.
1
Intro
5
Trær
3
D&C
4
Rang, O(n)
11
APSP
6
DP
13
NPC 1
8
Travers.
10
SSSP
7
Grådighet
14
NPC 2
12
Flyt
9
MST