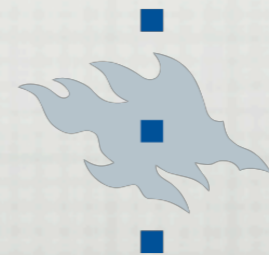


JOHDATUS TEKOÄLYYN

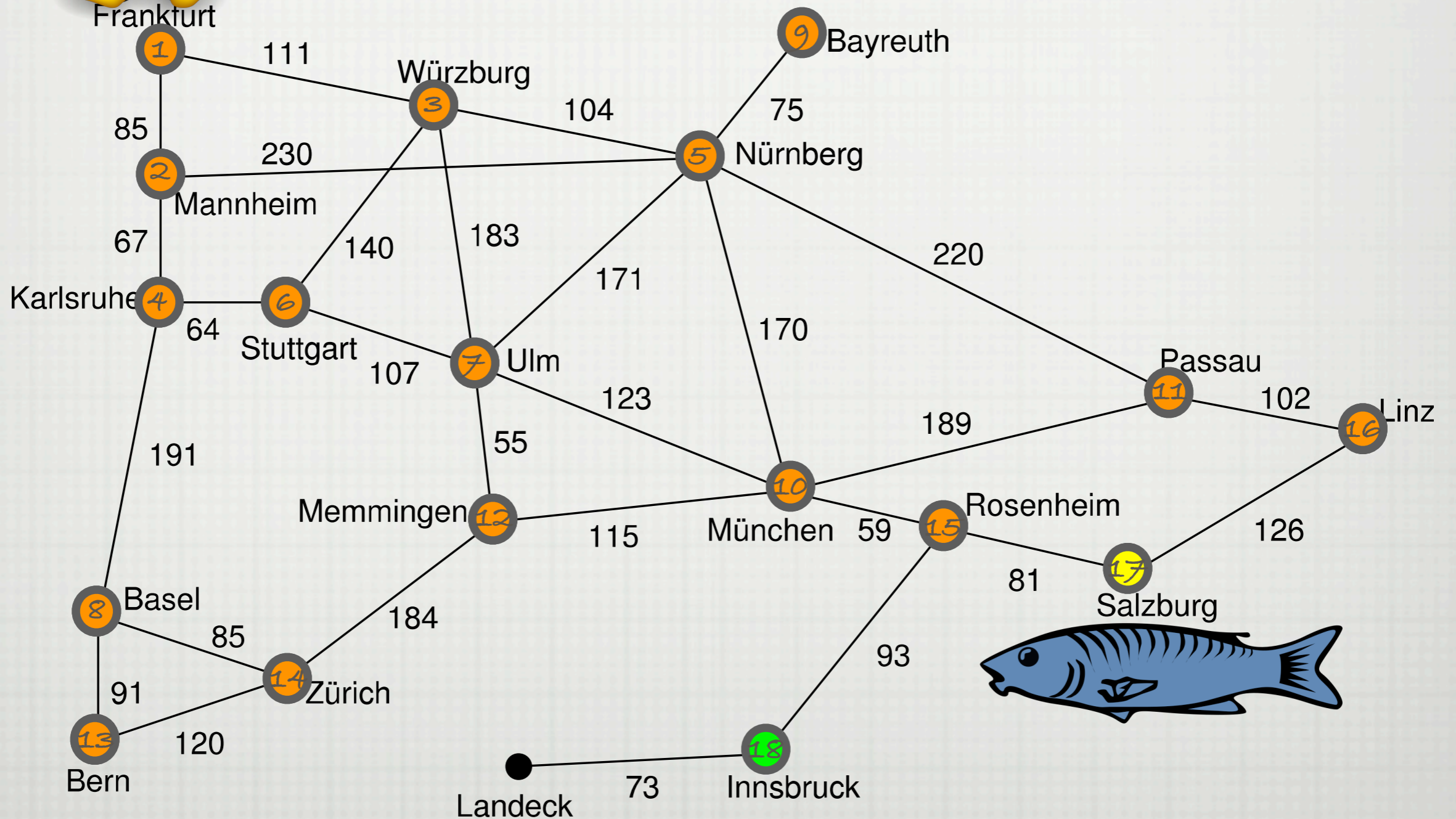
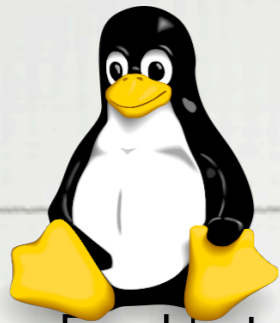
TEEMU ROOS



HELSINGIN YLIOPISTO

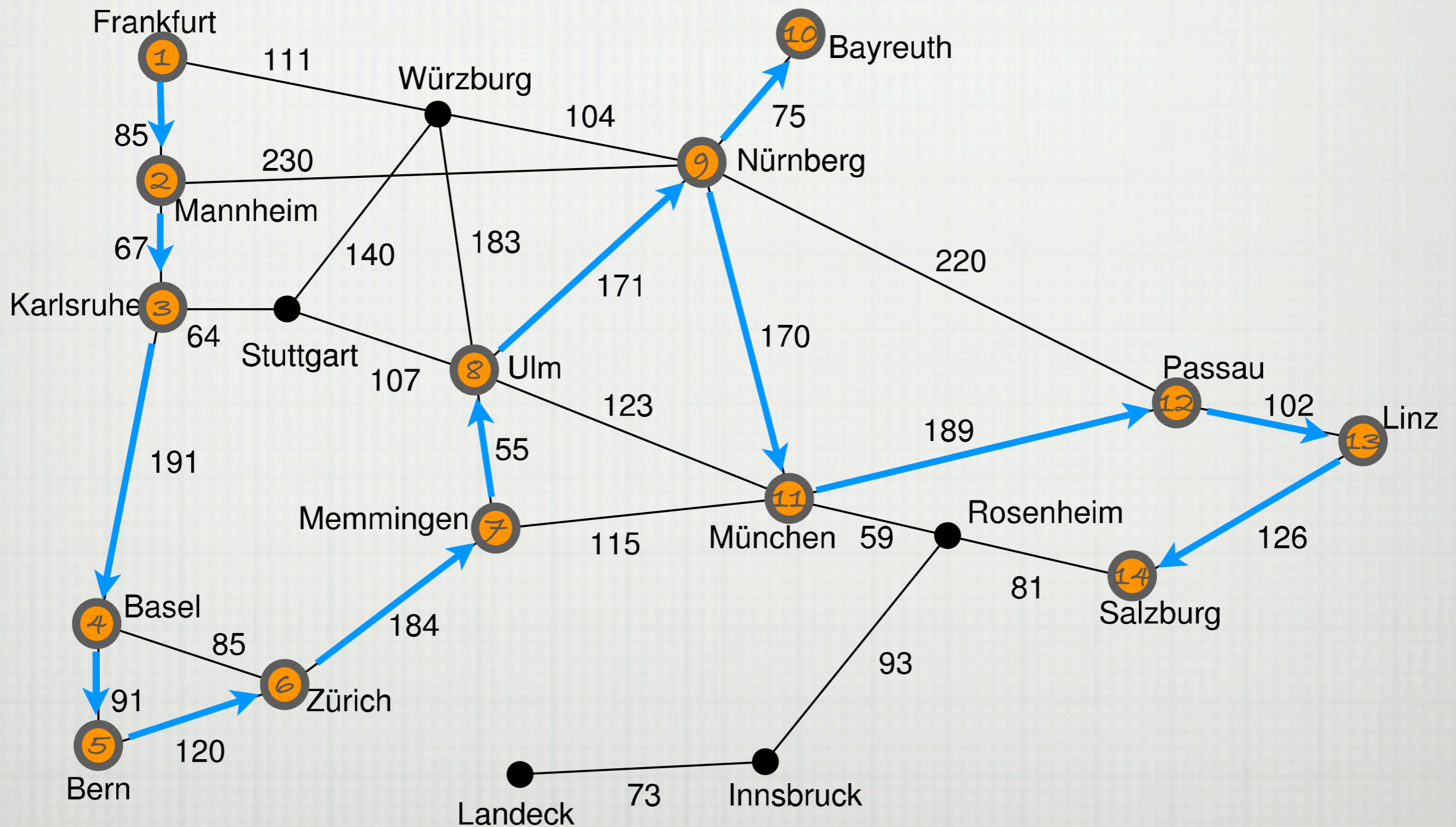
2. ETSINTÄ JA PELIT

LEVEYSSUUNTAINEN HAKU



2. ETSINTÄ JA PELIT

SYVYYSSUUNTAINEN HAKU



2. ETSINTÄ JA PELIT

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")

LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

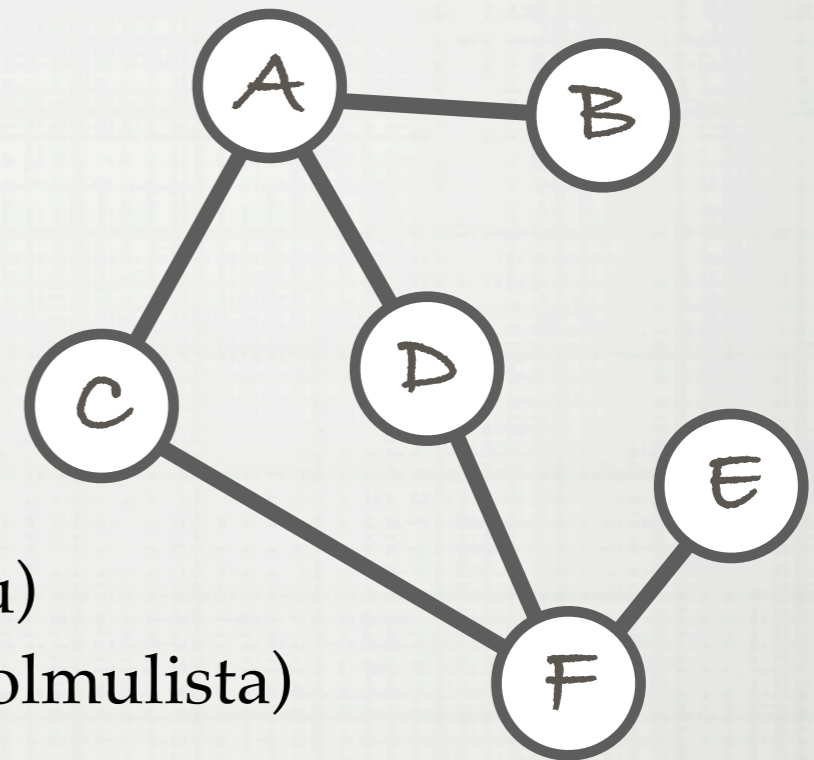
 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[AI]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

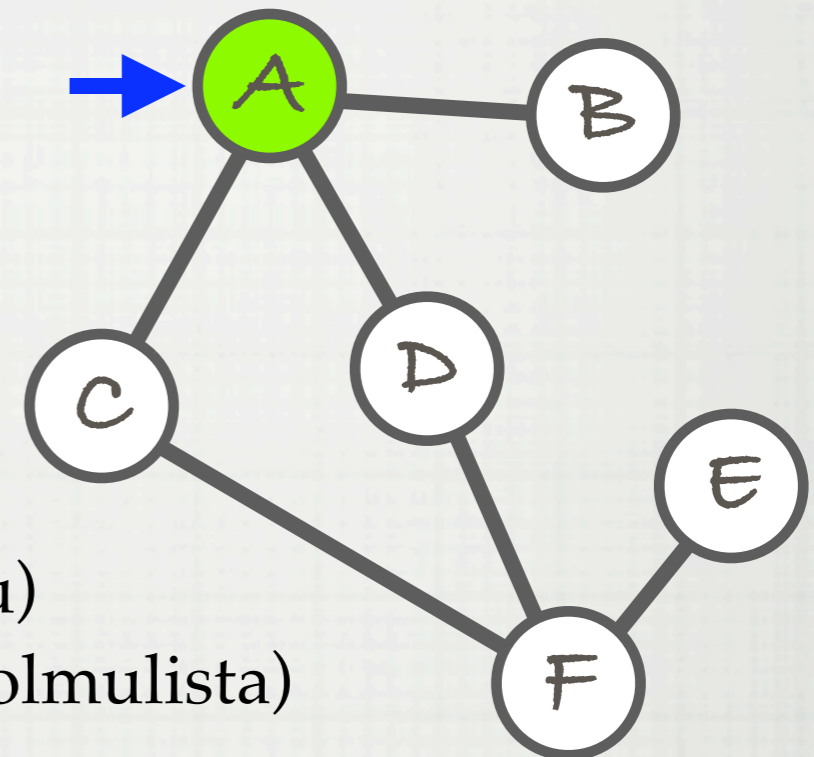
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT



ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

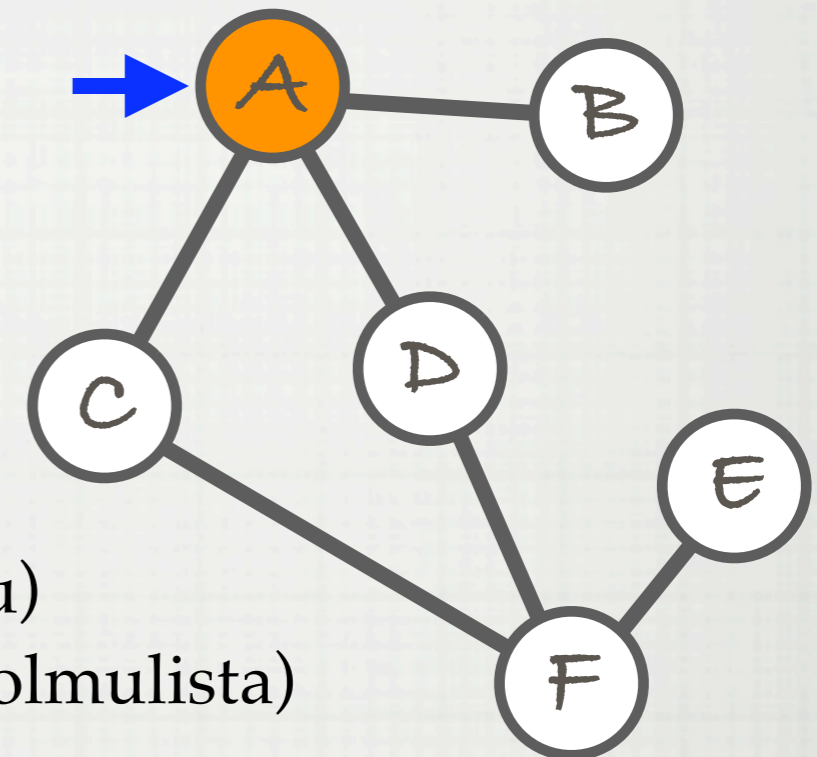
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[B,C,D]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

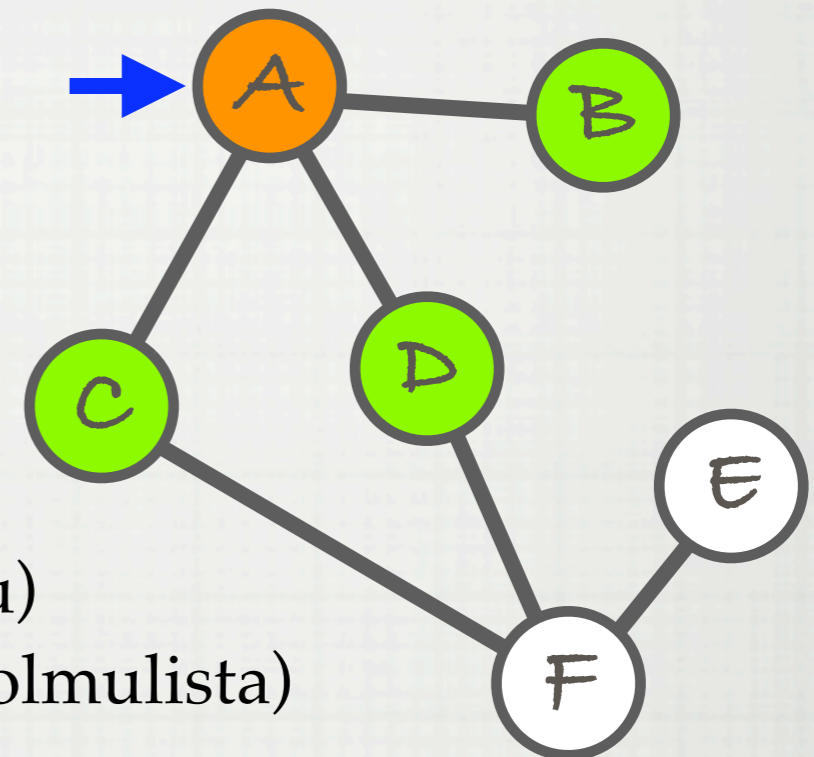
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[C,D]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

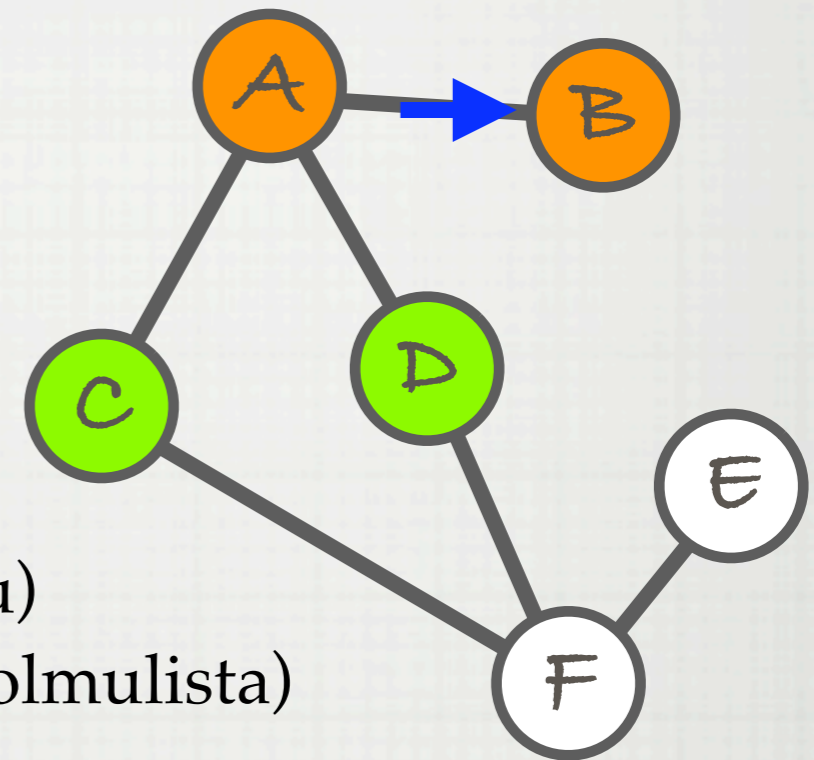
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[D]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

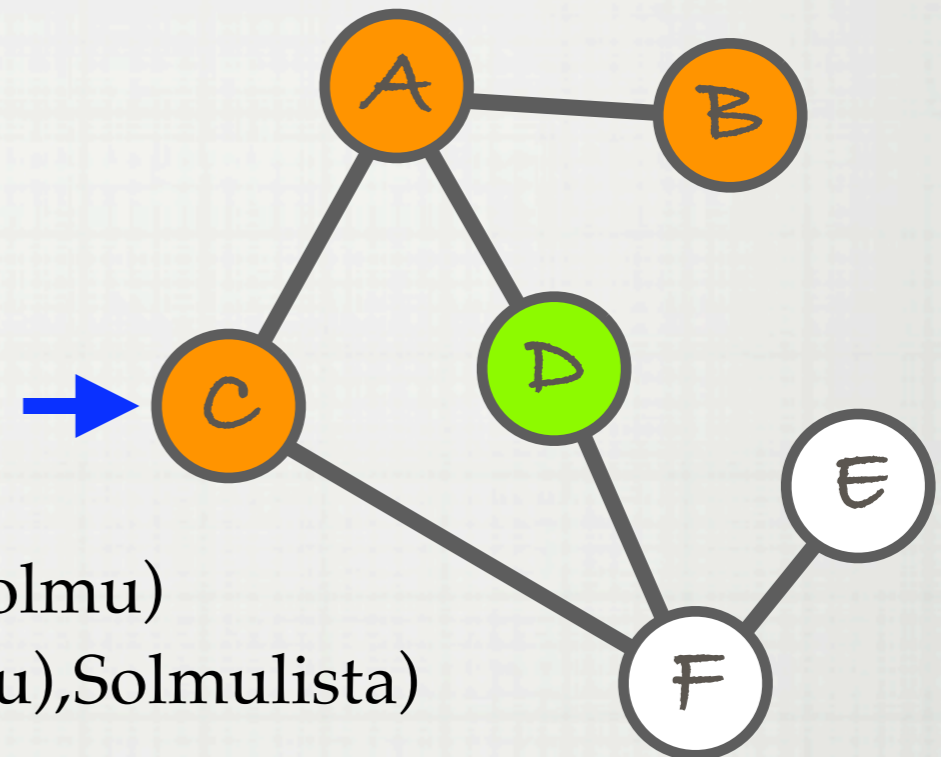
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[D,F]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

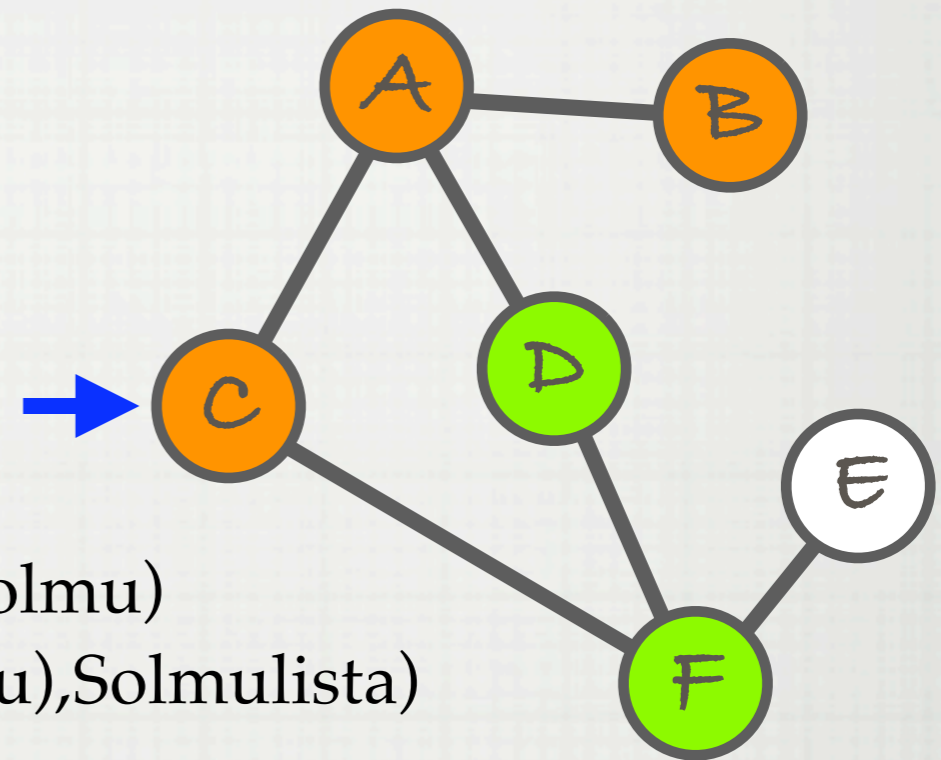
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[F]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

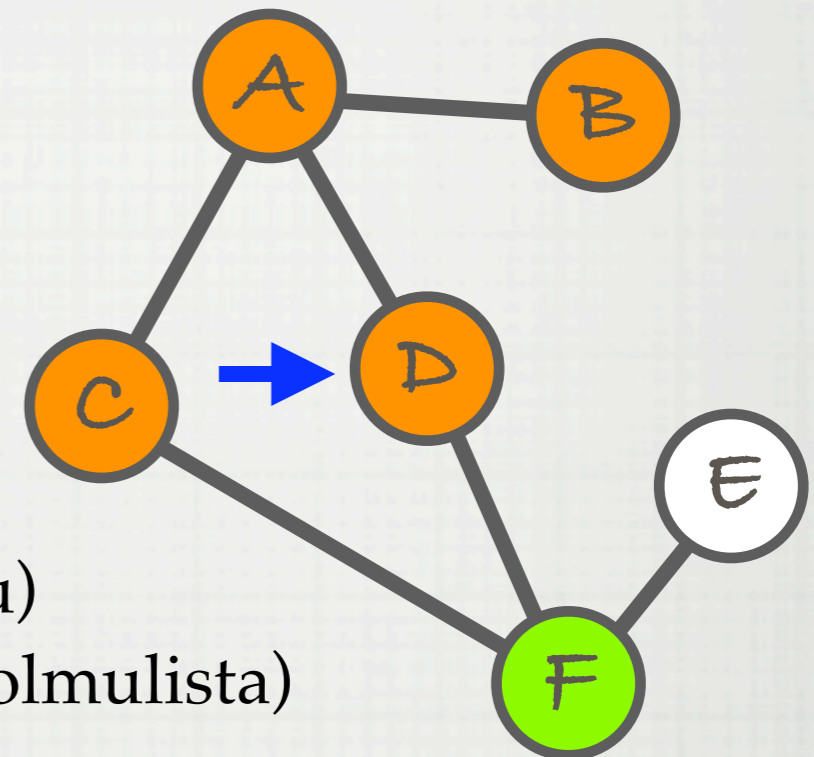
 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT



ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

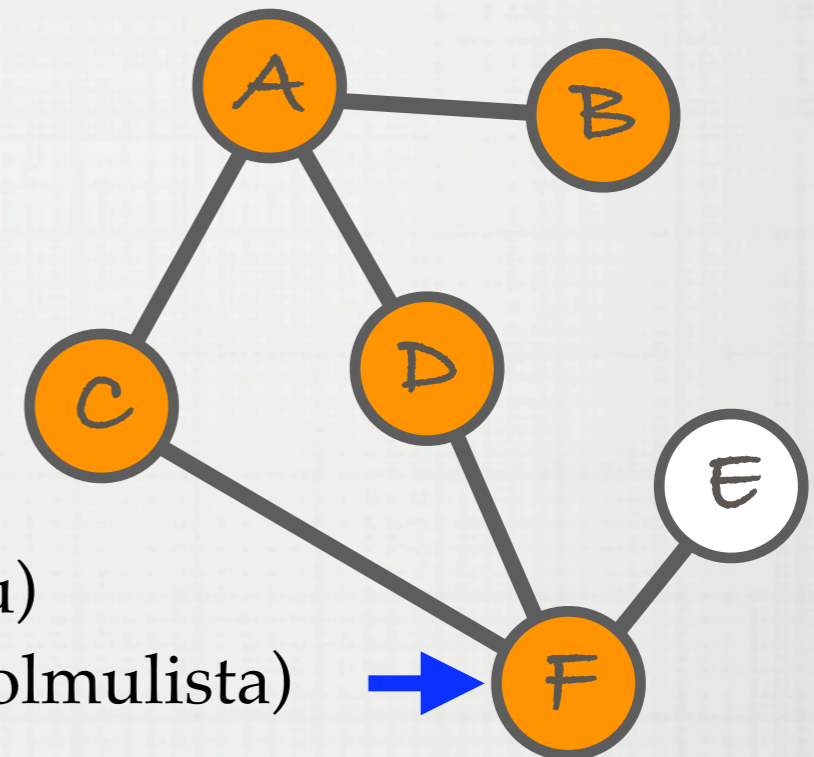
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

[E]

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

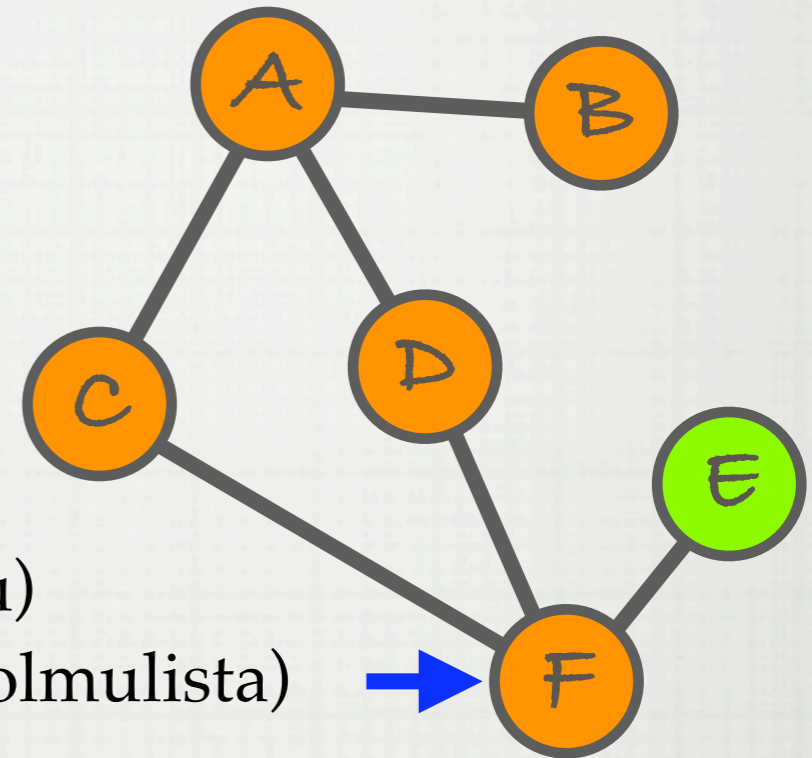
 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT



ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

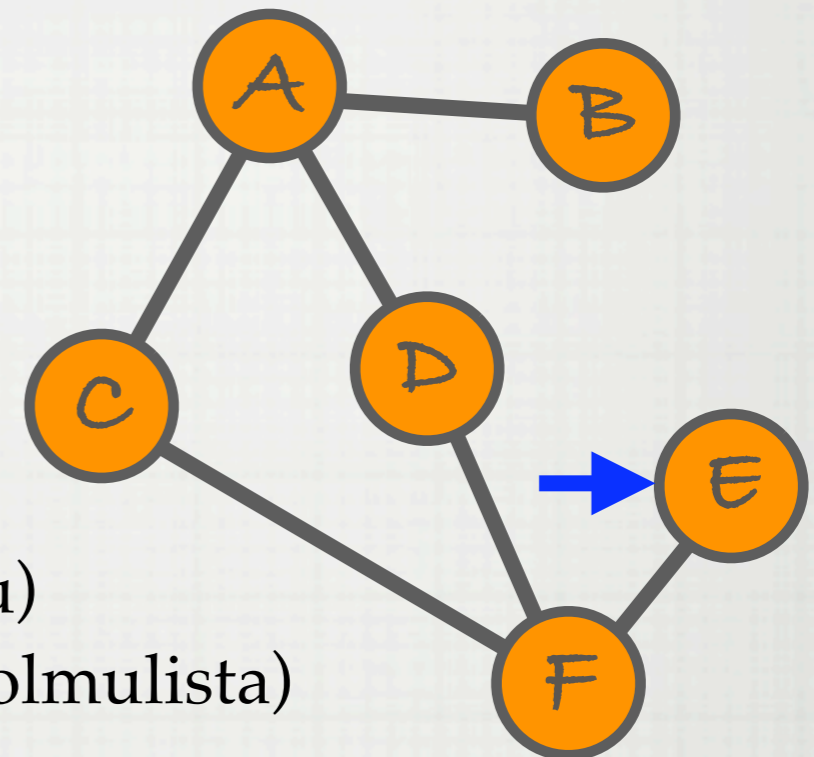
Solmu = EKA(Solmulista)

Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

Solmulista = LISÄÄ(NAAPURIT(Solmu),Solmulista)

return("ei ratkaisua")



LEVEYSSUUNTAINEN HAKU:

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

2. ETSINTÄ JA PELIT

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPI(Solmu), Solmulista)

return("ei ratkaisua")

LEVEYSSUUNTAINE

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(Solmulista2, UUDET(Solmulista1)))

LISÄÄ([a,f],[d]) => [d,a,f]

JONO ("FIRST-IN-FIRST-OUT")

2. ETSINTÄ JA PELIT

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

 Solmu = EKA(Solmulista)

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu), Solmulista)

return("ei ratkaisua")

SYVYYSSUUNTAINE

LISÄÄ(Solmulista1, Solmulista2)

return(PERÄKKÄIN(UUDET(Solmulista1), Solmulista2))

LISÄÄ([a,f],[d]) => [a,f,d]

PINO ("LAST-IN-FIRST-OUT")

ETSINTÄ ONGELMANRATKAISUNA



ETSINTÄ ONGELMANRATKAISUNA

Log In / Register

Send to a Friend

Put in Your Site

Missionaries and Cannibals

START
High Scores
more games



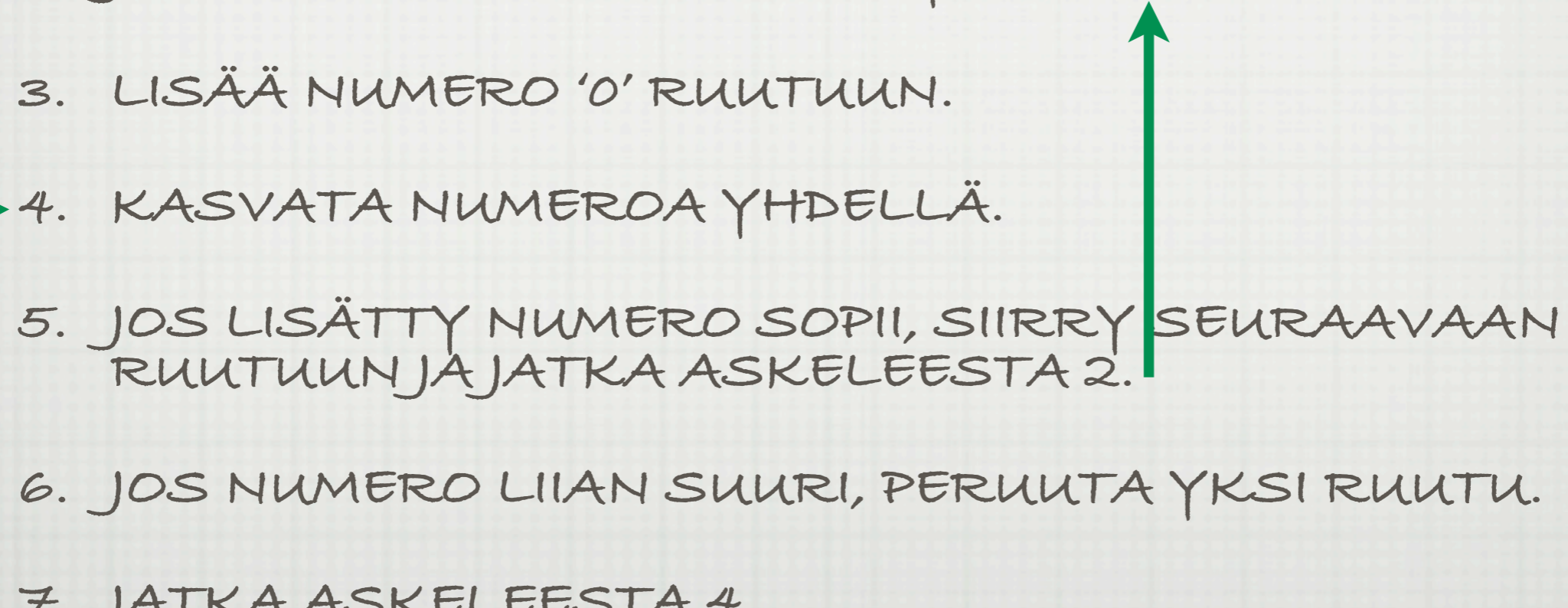
© Copyright, 2007, Novel Games Limited. All Rights Reserved.

ETSINTÄ ONGELMANRATKAISUNA

- * KOLME KANNIBAALIA JA KOLME LÄHETYSSAARNAAJAA HALUAA YLITTÄÄ JOEN VENEELLÄ, JOHON MAHTUU VAIN KAKSI HENKILÖÄ.
- * JOS JOMMALLA KUMMALLA RANNALLA ON ENEMMÄN KANNIBAALIA KUIN LÄHETYSSAARNAAJIA (MUTTA KUITENKIN VÄHINTÄÄN YKSI LÄHETYSSAARNAAJA), KANNIBAALIT SYÖVÄT HEIDÄT.
- * MITEN JOKI SAADAAN YLITETTYÄ ILMAN, ETTÄ KETÄÄN SYÖDÄÄN?
- * VOIT KOKEILLA KLIKKAAMALLA TÄSTÄ.

SUDOKU

* YKSINKERTAINEN SUDOKU-ALGORITMI:

1. ALOITA VASEMMASTA YLÄKULMASTA.
 2. JOS RUUTU ANNETTU, SIIRRY SEURAAVAAN.
 3. LISÄÄ NUMERO '0' RUUTUUN.
 4. KASVATA NUMEROA YHDELLÄ.
 5. JOS LISÄTTY NUMERO SOPII, SIIRRY SEURAAVAAN RUUTUUN JA JATKA ASKELEESTA 2.
 6. JOS NUMERO LIIAN SUURI, PERUUTA YKSI RUUTU.
 7. JATKA ASKELEESTA 4.
- 

SUDOKU

	3		
	2		1
	1	2	3

SUDOKU

1	3		
	2		1
	1	2	3

SUDOKU

1	3	1	
	2		1
	1	2	3

SUDOKU

1	3	2	
	2		1
	1	2	3

SUDOKU

1	3	3	
	2		1
	1	2	3

SUDOKU

1	3	4	
	2		1
	1	2	3

SUDOKU

1	3	4	2
	2		1
	1	2	3

SUDOKU

1	3	4	2
2			
	2		1
	1	2	3

SUDOKU

1	3	4	2
2	4		
	2		1
	1	2	3

SUDOKU

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

SUDOKU

1	3	4	2
2	4	1	?
	2		1
	1	2	3

SUDOKU

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

SUDOKU

1	3	4	2
2	4	3	?
	2		1
	1	2	3

SUDOKU

1	3	4	2
2	4	?	
	2		1
	1	2	3

SUDOKU

1	3	4	2
2	?		
	2		1
	1	2	3

SUDOKU

1	3	4	2
4			
	2		1
	1	2	3

SUDOKU

1	3	4	2
4	?		
	2		1
	1	2	3

SUDOKU

1	3	4	2
?			
	2		1
	1	2	3

SUDOKU

1	3	4	?
	2		1
	1	2	3

SUDOKU

1	3	?	
	2		1
	1	2	3

SUDOKU

2	3		
	2		1
	1	2	3

SUDOKU

2	3	1	
	2		1
	1	2	3

SUDOKU

2	3	1	4
	2		1
	1	2	3

SUDOKU

2	3	1	4
1			
	2		1
	1	2	3

SUDOKU

2	3	1	4
1	4		
	2		1
	1	2	3

SUDOKU

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

SUDOKU

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

SUDOKU

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

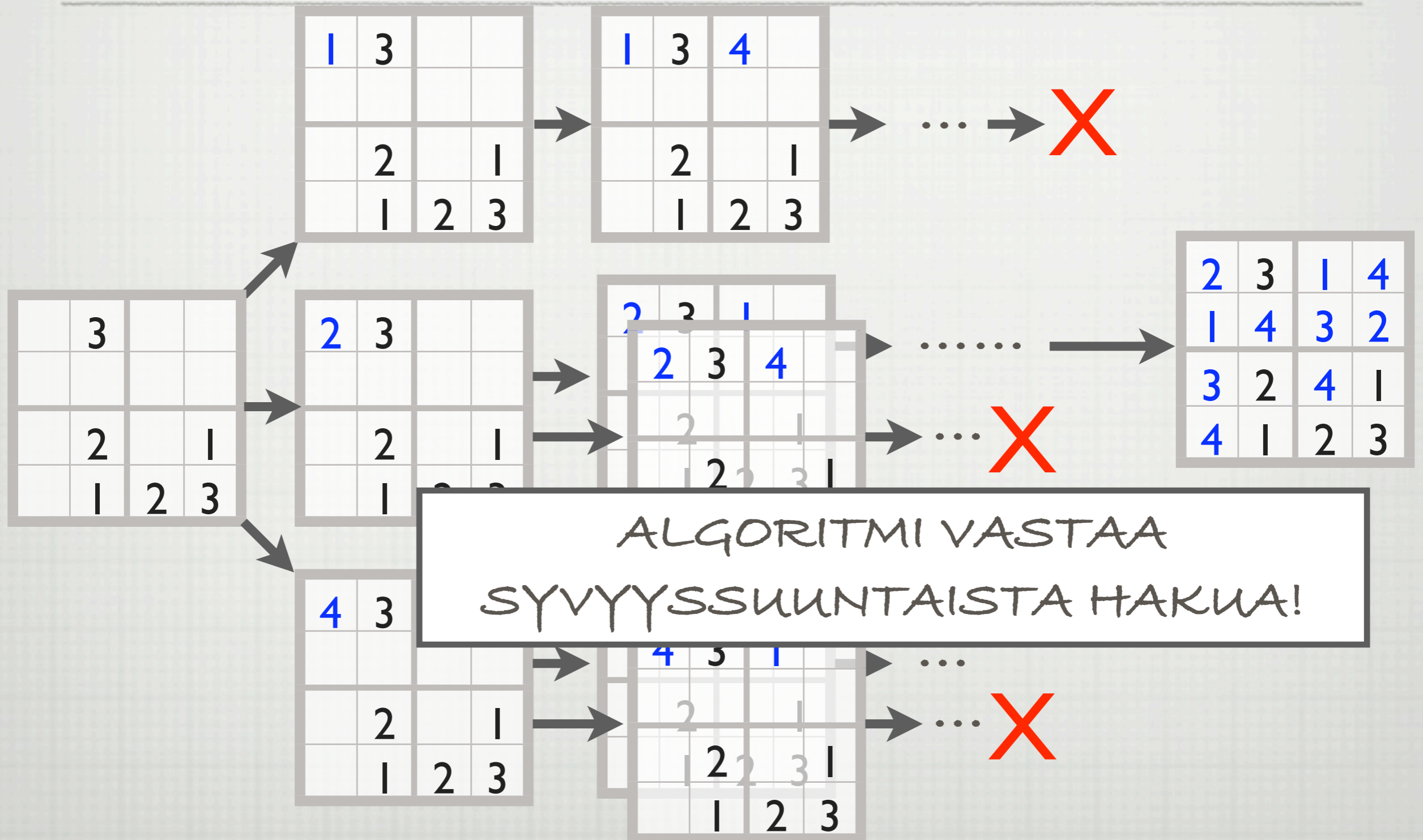
SUDOKU

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

SUDOKU

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

SUDOKU



ALGORITMI VASTAA
SYVYYSSUUNTAISTA HAKUA!

HEURISTIIKAT

- * **KUSTANNUSARVIO:** $f(N)$
 - ARVIO SOLMUN N KAUTTA KULKEVAN POLUN KUSTANNUKSESTA
- * **"HEURISTIIKKA":** $h(N)$
 - ARVIO SOLMUSTA N LÄHTEVÄN POLUN KUSTANNUKSESTA
- * **POLKUKUSTANNUS:** $g(N)$
 - SOLMUUN N PÄÄTTYVÄN POLUN KUSTANNUS

$$f(N) = g(N) + h(N)$$

HEURISTINEN HAKU

ETSINTÄ(Alkusolmu)

Solmulista = [Alkusolmu]

while Solmulista ei tyhjä

 Solmu = EKA(Solmulista);

 Solmulista = LOPUT(Solmulista)

if MAALI(Solmu) **return**("ratkaisu", Solmu)

 Solmulista = LISÄÄ(NAAPURIT(Solmu), Solmulista)

return("ei ratkaisua")

PARAS-ENSIN-HAKU

LISÄÄ(Solmulista1, Solmulista2)

return(JÄRJESTÄ(Solmulista1, Solmulista2))

[(a,5),(b,3),(c,1)], [(a,2),(c,3),(f,5)] => [(c,1),(a,2),(b,3),(f,5)]

A*-HAKU: $f(N) = g(N) + h(N)$

A*

- * **OLETUS:** HEURISTIikka $h(N)$ ANTAA AINA ENINTÄÄN YHTÄ SUUREN ARVON KUIN TODELLINEN KUSTANNUS SOLMUSTA N MAALIIN.
- * TÄLLÖIN **A*** TUOTTAA AINA OPTIMAALISEN RATKAISUN
- * TODISTUKSEN IDEA:
JONON EKAKSI EI VOI PÄÄSTÄ MAALISOLMU, JONKA KUSTANNUS ON SUUREMPI KUIN OPTIMAALISEN MAALISOLMUN.
- * JOS HEURISTIikka "HYVÄ", SUURIMMASSA OSASSA HUONOJA SOLMUJA EI KÄYDÄ OLLENKAAN.

REITTIOPAS

[Taskuversio](#) • [På svenska](#) • [In English](#) • [Slangi](#) • [По-русски](#)

[Palaute](#) • [Ohjeet](#) • [FAQ](#)



Reittiopas

[Reittiopas classic](#) • [Reittiopas API](#)

HSL

Reittiopas

[Omat lähdöt](#) • [Aikataulut](#) • [Linjaopas](#) • [Pyöräily ja kävely](#)



Perushaku

Tarkennettu haku

Mistä [Kartta](#) [Tallenna](#) [Hakemisto](#)

Mihin [Kartta](#) [Tallenna](#) [Hakemisto](#)

Kello : Lähtöaika
 Perillä

Pvm

	Ma	Ti	Ke	To	Pe	La	Su	
36	05	06	07	08	09	10	11	
Syyskuu 37	12	13	14	15	16	17	18	
2011 38	19	20	21	22	23	24	25	
39	26	27	28	29	30	01	02	
Lokakuu 40	03	04	05	06	07	08	09	

Hae

Poikkeusinfo



8.9.2011 17:35 - Sisäiset ja seutuliikenne myöhästyvät. Syy: ruuhka. Paikka: Mannerheimintie. Arvioitu kesto: 17:31 - 24:00.



8.9.2011 17:32 - Sisäiset ja seutulinjat myöhästyvät. Syy: ruuhka. Paikka: Mannerheimintie. Arvioitu kesto: 17:23 - 24:00.

[HSL /Poikkeusinfo](#) → [Mobiililaitteille](#) →

Liikennetiedotteet

- 8.9.2011 [Linjalla 11 lisävuoroja Kissojen yöhön 9.9.](#)
- 8.9.2011 [Martinkyläntien pysäkkien poikkeusjärjestelyt jatkuvat](#)
- 7.9.2011 [Bussille 14 reittimuutos Eirassa 12.9.](#)
- 7.9.2011 [Bussit ajavat poikkeusreittiä Keravan keskustassa 9. - 12.9.](#)

Omat reitit

Ei omia reittejä ([ohje omien reittien tallentamiseen](#)).

Omat paikat

Ei omia paikkoja ([ohje omien paikkojen tallentamiseen](#)).

REITTIOPAS

- * TILA: (PYSÄKKI, KLO)
- * SIIRTYMÄT: (LINJANRO, SUUNTA)
- * TEHTÄVÄ: ETSI NOPEIN REITTI PYSÄKILTÄ A PYSÄKILLE B (EI KÄVELYÄ)
- * MENETELMÄ: A*-HAKU