Aalto-yliopisto

Laskennan vaativuus ja NP-täydelliset ongelmat

TRAK-vierailuluento 13.4.2010

Petteri Kaski
Tietojenkäsittelytieteen laitos

Tietojenkäsittelytiede

- Tietojenkäsittelytiede tutkii
  1. mitä tehtäviä voidaan automatisoida, ja
  2. miten tämä tehdään *tehokkaasti*

- Kuten tieteessä yleensä, tietojenkäsittelytieteessä eräänä keskeisenä tavoitteena on *luokitella* tarkasteltavaa aineistoa

- Aineisto:

Laskennalliset ongelmat

Lajittelu

- Syöte:
  Taulukko *a*[1],*a*[2],...,*a*[*n*] kokonaislukuja
- Tehtävä:
  Lajittele taulukko kasvavaan järjestykseen

Toistuvat alkiot (päätös)

- Syöte:
  Taulukko *a*[1],*a*[2],...,*a*[*n*] kokonaislukuja
- Tehtävä:
  Onko taulukossa toistuvia alkioita? (kyllä/ei)

Mediaani

- Syöte:
  Taulukko *a*[1],*a*[2],...,*a*[*n*] kokonaislukuja
- Tehtävä:
  Laske taulukon mediaani

Maksimi

- Syöte:
  Taulukko *a*[1],*a*[2],...,*a*[*n*] kokonaislukuja
- Tehtävä:
  Määritä taulukon suurin arvo

- Kuinka tehokkaasti osaamme ratkaista ongelman?
  - Esimerkiksi: algoritmin ajoaika syötteen koon (=*n*) funktiona
- Mitkä ovat ongelmien väliset suhteet?
  - Esimerkiksi: auttaako toisen ratkaisu toisessa?

Luennon sisältö

- Esimerkkejä laskennallisista ongelmista
- Laskennan vaativuus --- tehokas laskenta
- Ongelmaluokat P ja NP
- P=NP ?
- Luokan NP rakenteesta:
  NP-täydelliset ongelmat

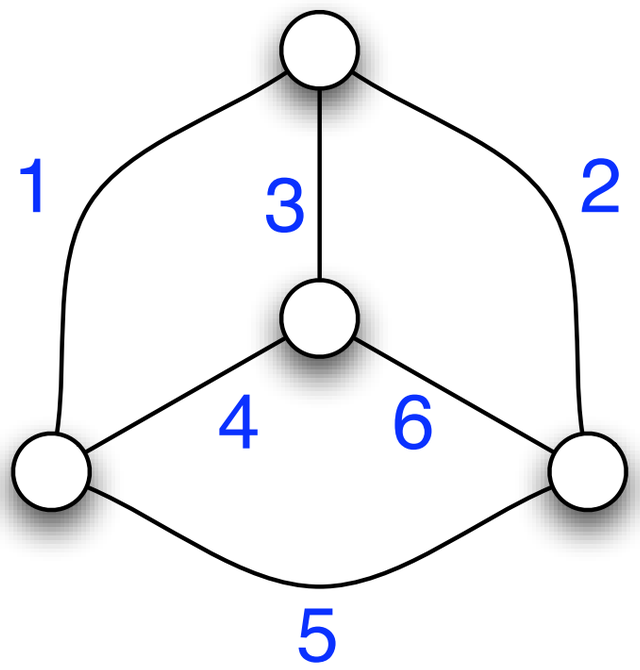
1. Esimerkkejä laskennallisista ongelmista

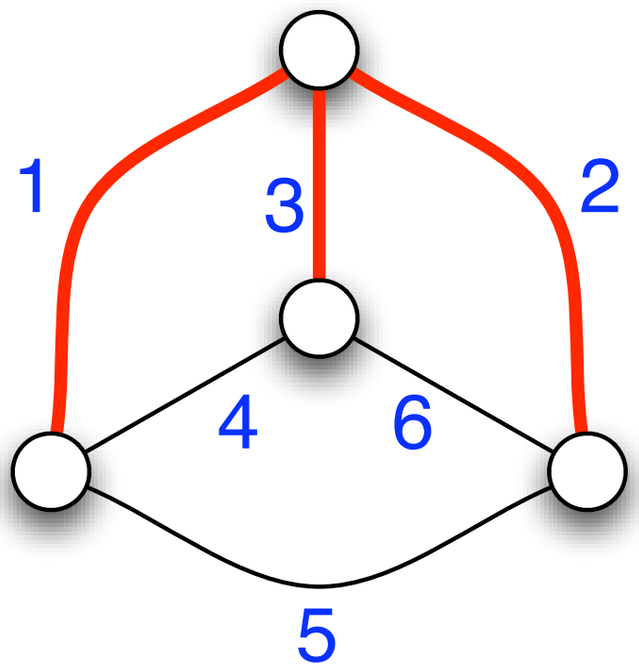
- Verkko-ongelmat
  - pienin virittävä puu
  - kauppamatkustajan ongelma
- Boolen logiikka
  - piirin arvo
  - piirin toteutuvuus
- Kokonaisluvut
  - kertolasku
  - tekijöinti
- Matriisien yhtäsuuruus
  - rivijärjestystä vaille
  - rivi- ja sarakejärjestystä vaille

Verkko-ongelmat

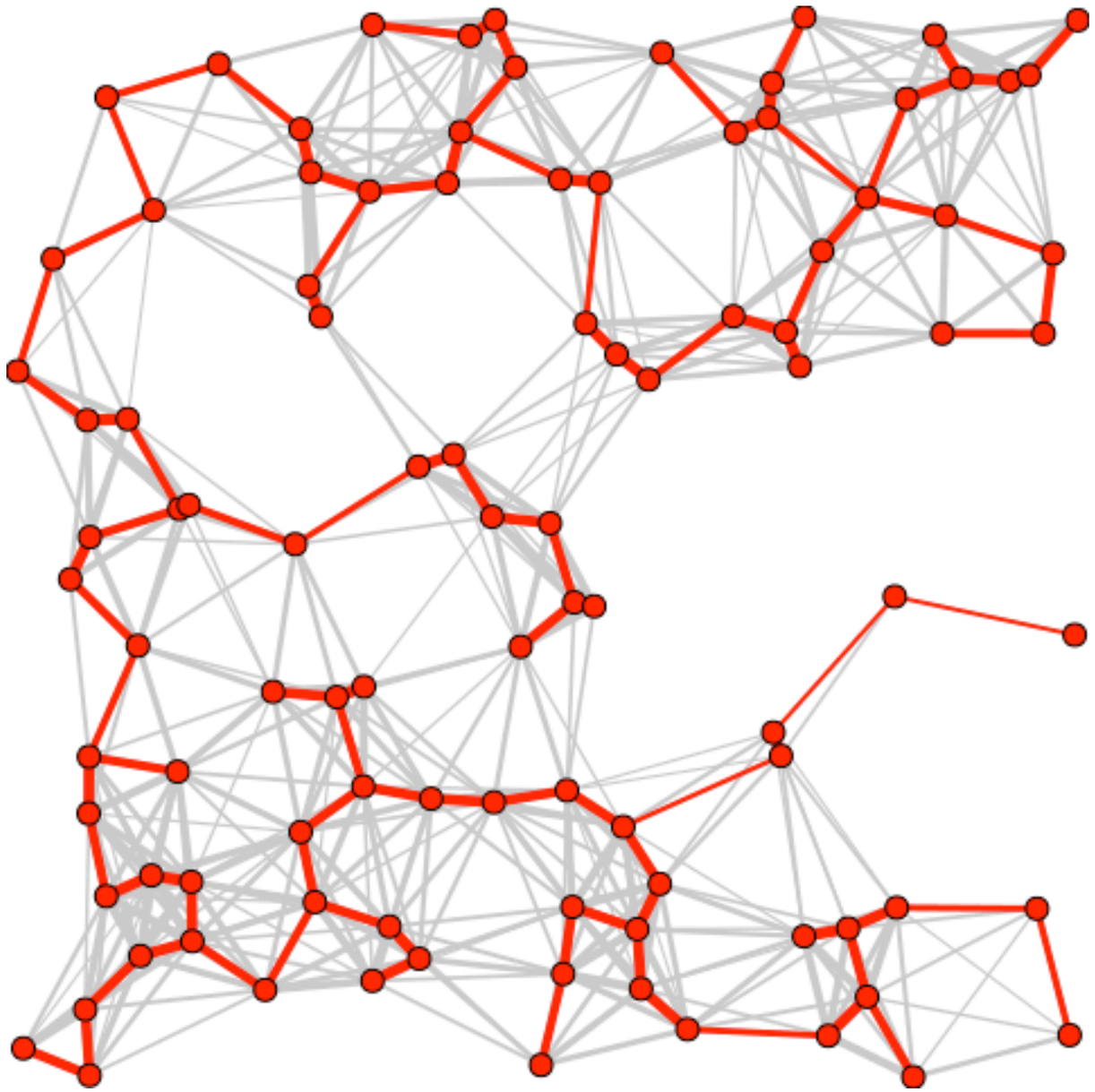
Pienin virittävä puu

- Syöte:
  Kaaripainotettu täydellinen verkko
- Tehtävä:
  Määritä verkolle virittävä puu, joka on kokonaispainoltaan pienin





Esimerkki:
Sähköverkon suunnittelu

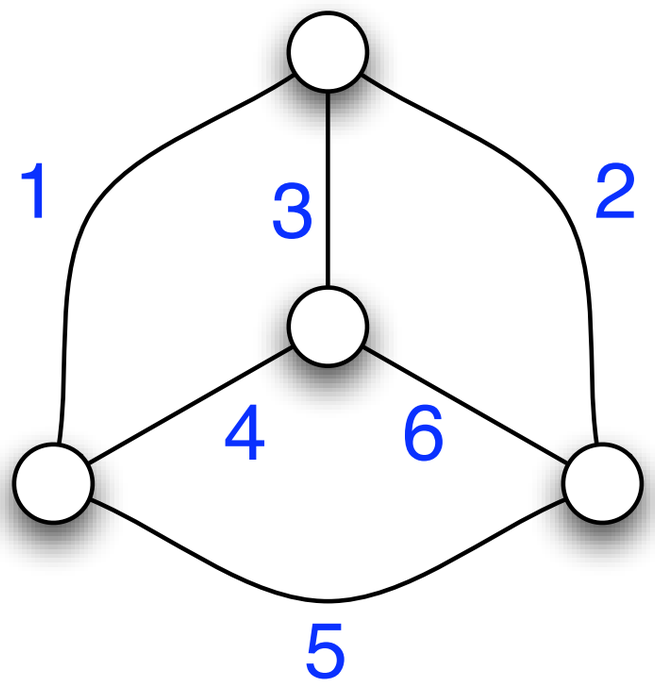


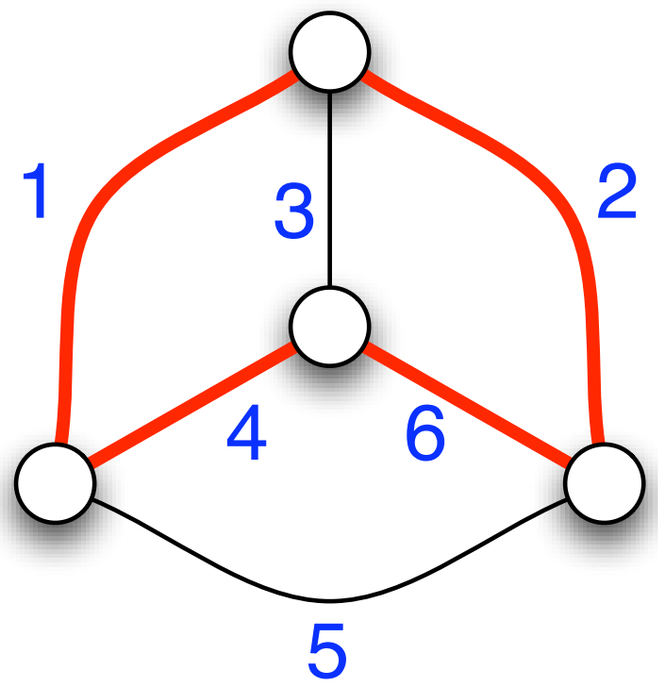
solmut =
sähköistettävät
kohteet

kaaripaino =
sähkölinjan
rakennuskustannus

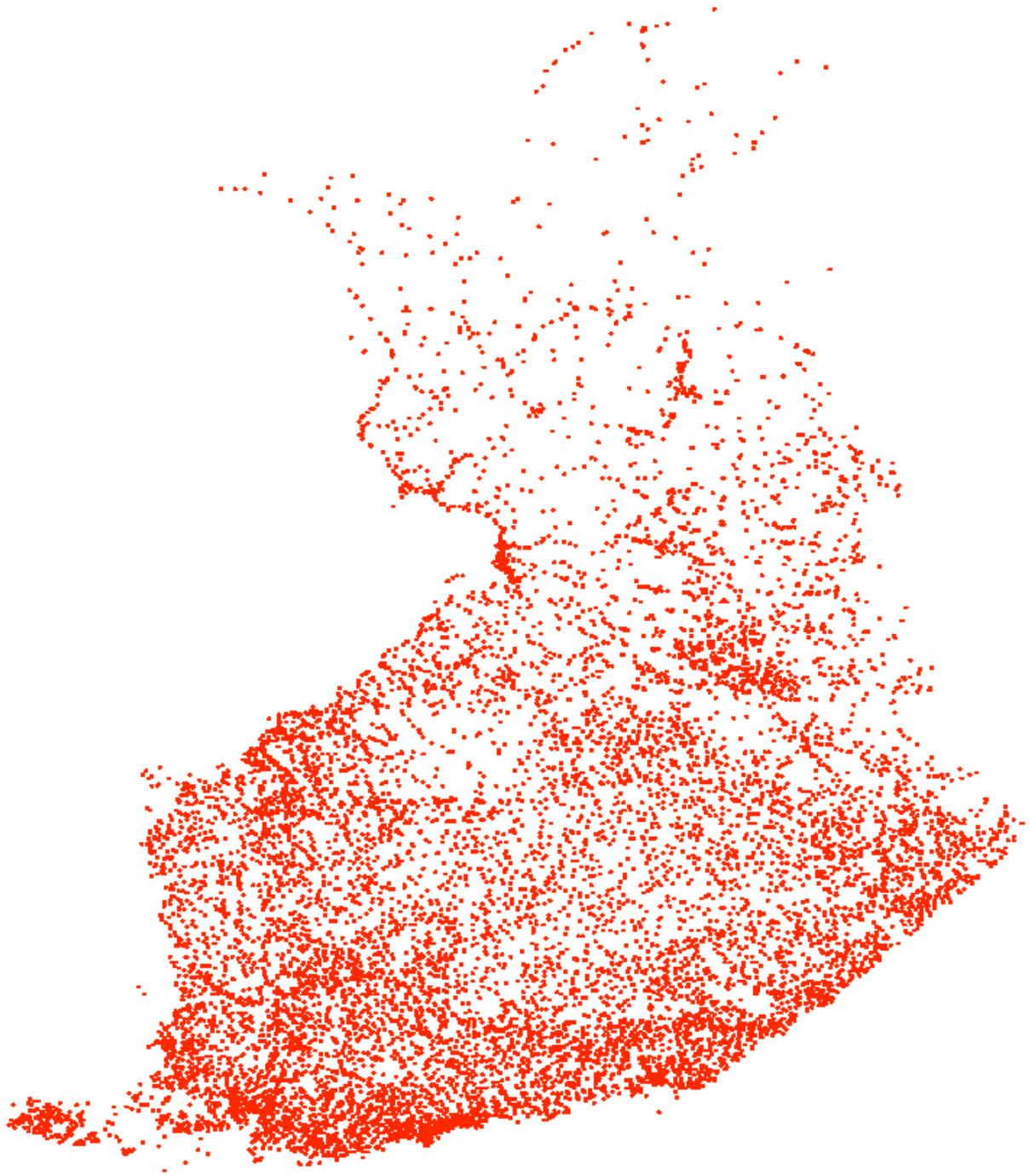
Kauppamatkustajan ongelma

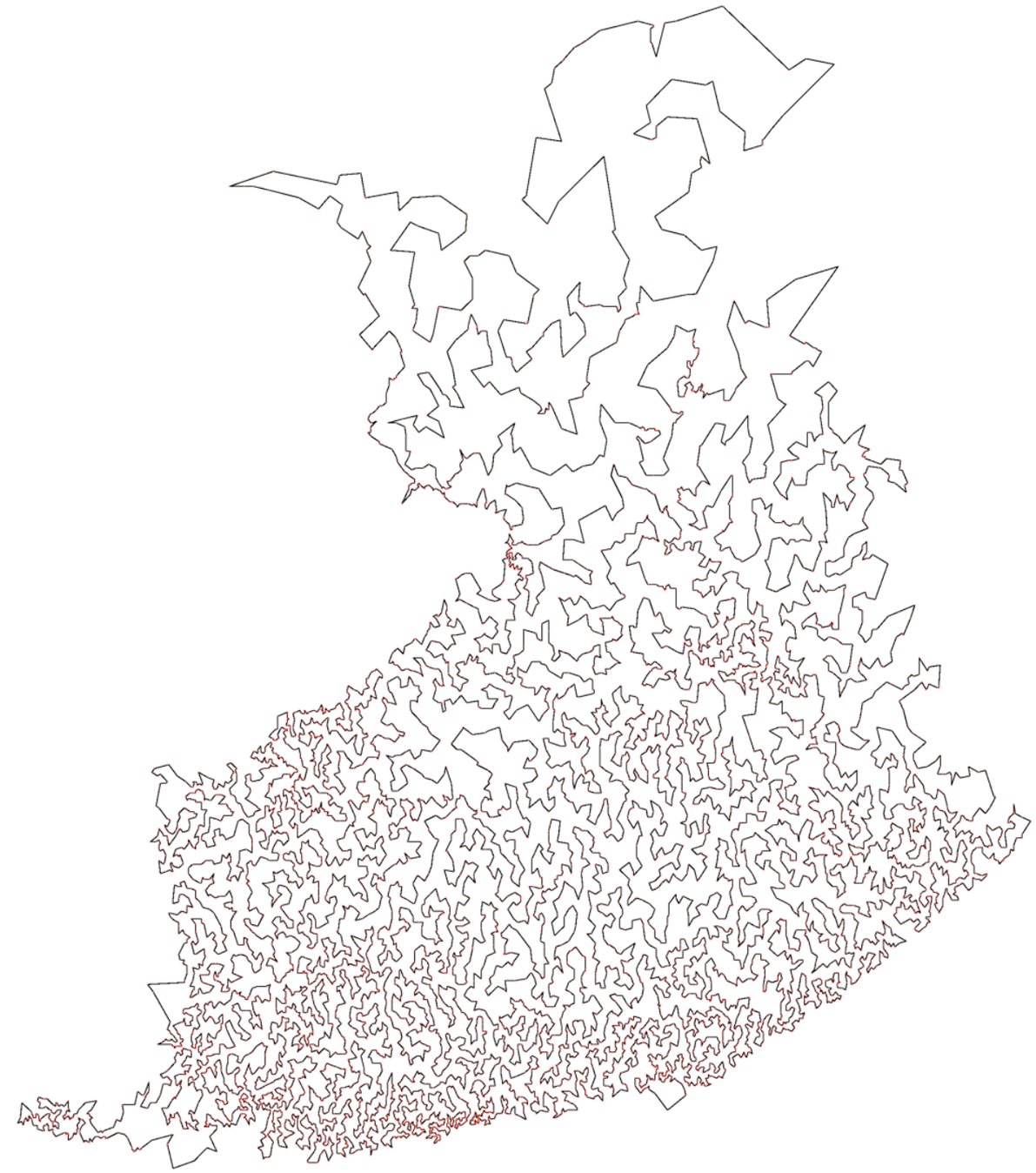
- Syöte:
  Kaaripainotettu täydellinen verkko
- Tehtävä:
  Määritä verkolle virittävä kehä, joka on kokonaispainoltaan pienin





10639 ‘populated locations’ in Finland

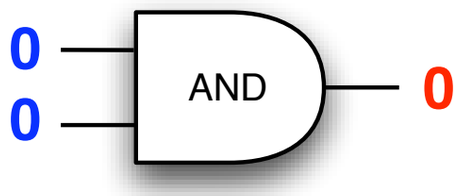


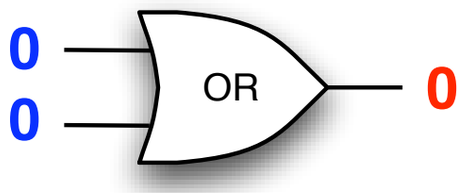


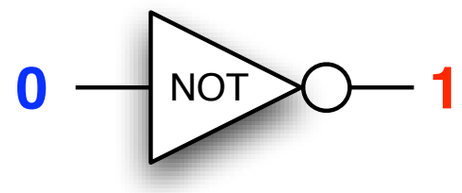
http://www.tsp.gatech.edu/

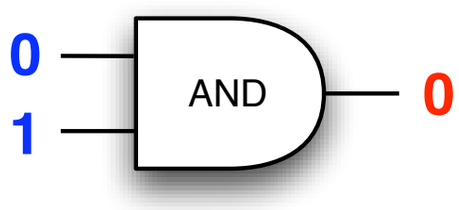
Boolen logiikka

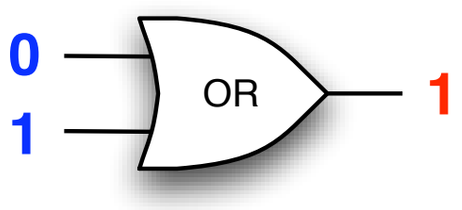
Logiikkaportit:
sisääntulo(t) ja ulostulo

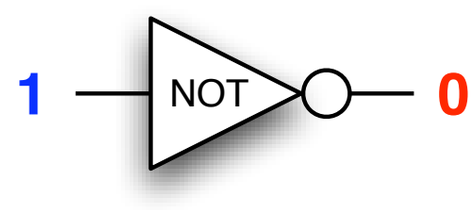


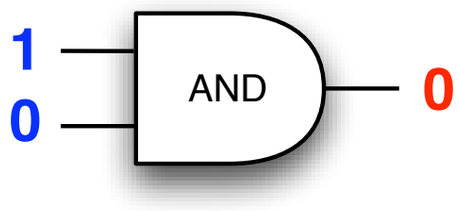


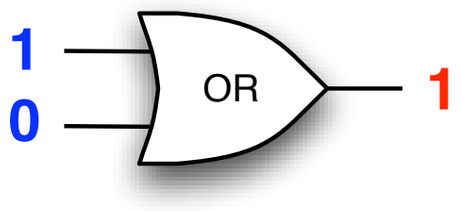


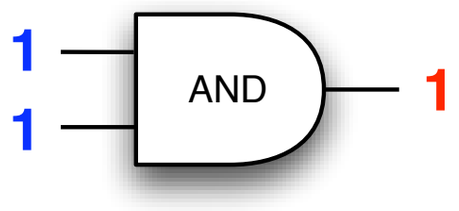


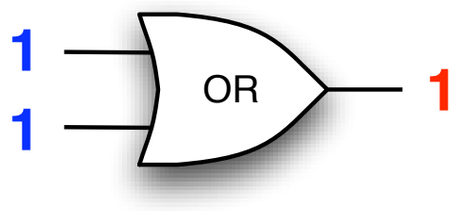










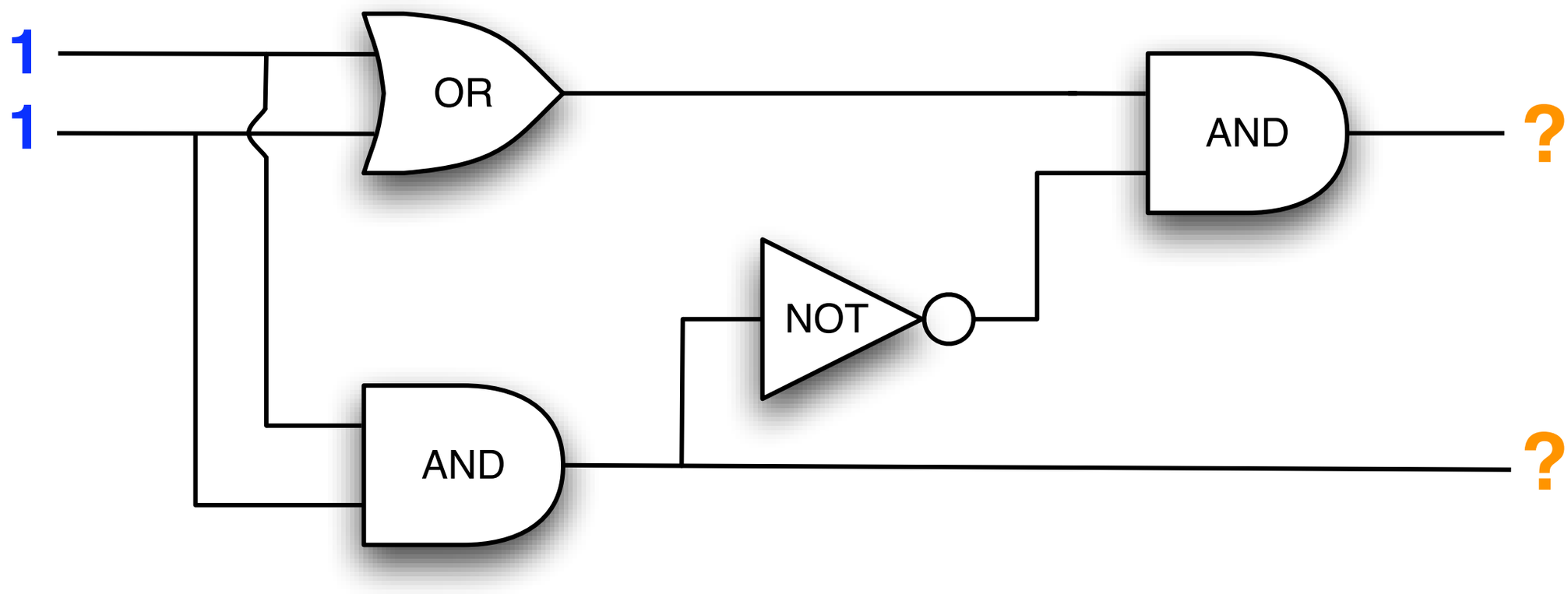


0 ~ epätosi

1 ~ tosi

Piirin arvo

- Syöte:
  Piiri ja arvot piirin sisääntuloille
- Tehtävä:
  Määritä arvot ulostuloissa



Esimerkki:
AES salausalgoritmi

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Federal Information

Processing Standards Publication 197

November 26, 2001

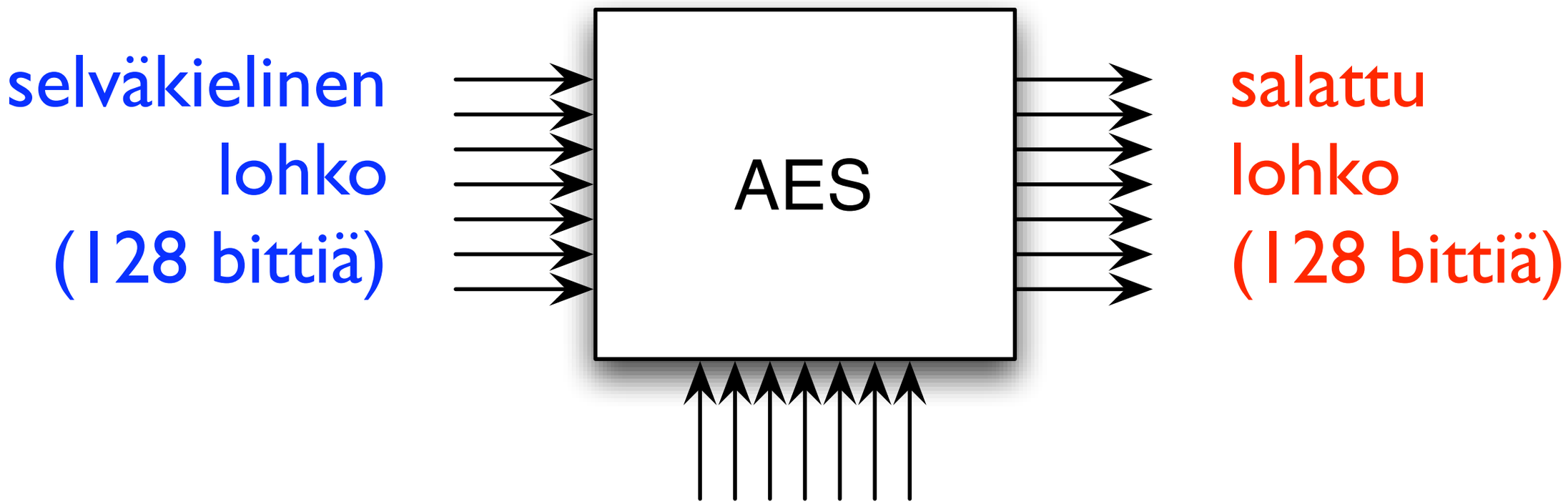
Announcing the

ADVANCED ENCRYPTION STANDARD (AES)

Federal Information Processing Standards Publications (FIPS PUBS) are issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce pursuant to Section 5131 of the Information Technology Management Reform Act of 1996 (Public Law 104-106) and the Computer Security Act of 1987 (Public Law 100-235).

1. **Name of Standard.** Advanced Encryption Standard (AES) (FIPS PUB 197).
2. **Category of Standard.** Computer Security Standard, Cryptography.

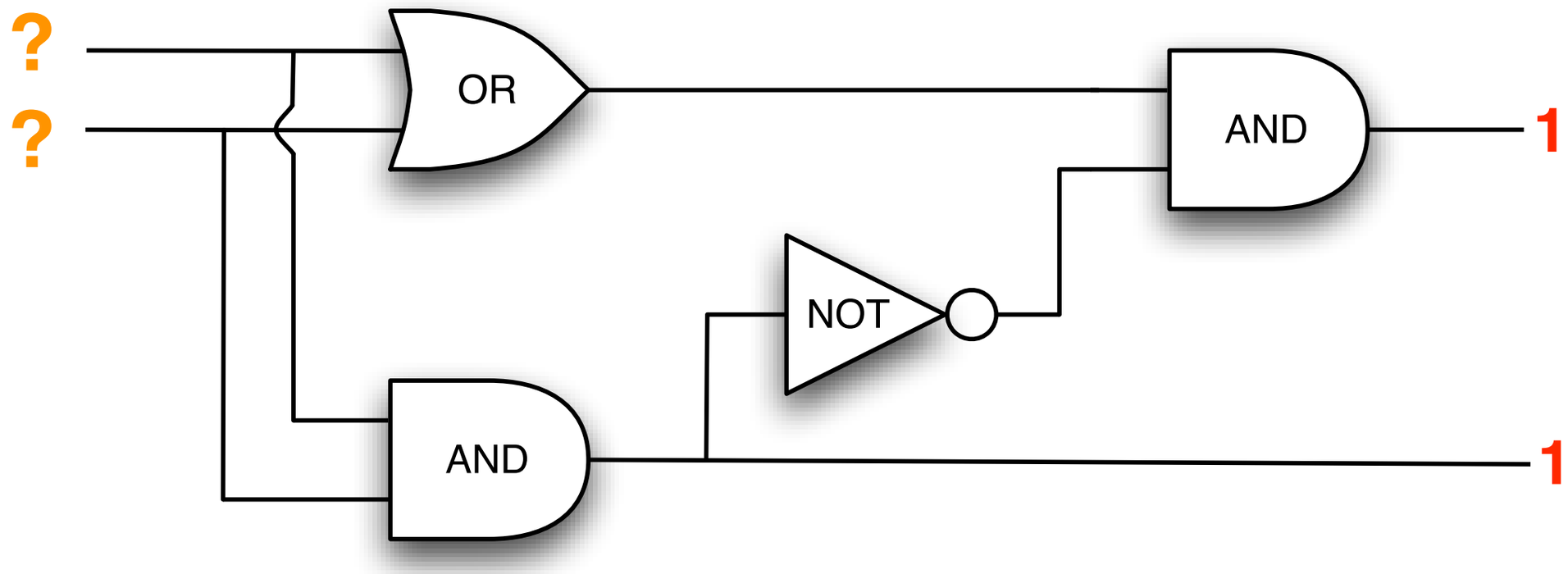
Esimerkki:
AES salausalgoritmi



Piirin toteutuvuus (päätös)

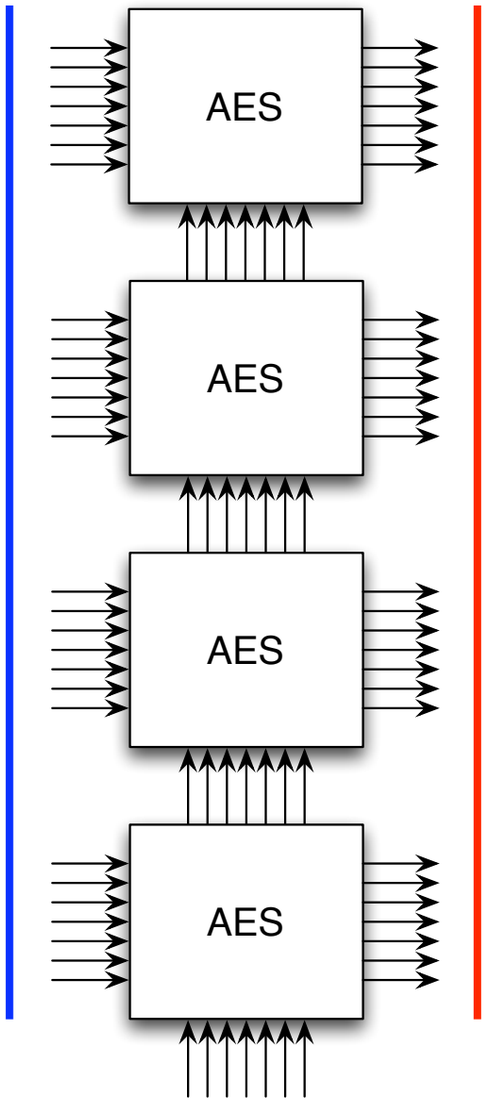
- Syöte:
  Piiri ja arvot piirin ulostuloille
  (sekä mahdollisesti osalle sisääntuloista)

- Tehtävä:
  Voidaanko tuntemattomille sisääntuloille asettaa arvot siten, että piiri saa halutut arvot? (kyllä/ei)



Esimerkki:
AES salausavaimen päättely

selväkielinen sanoma (tunnettu)

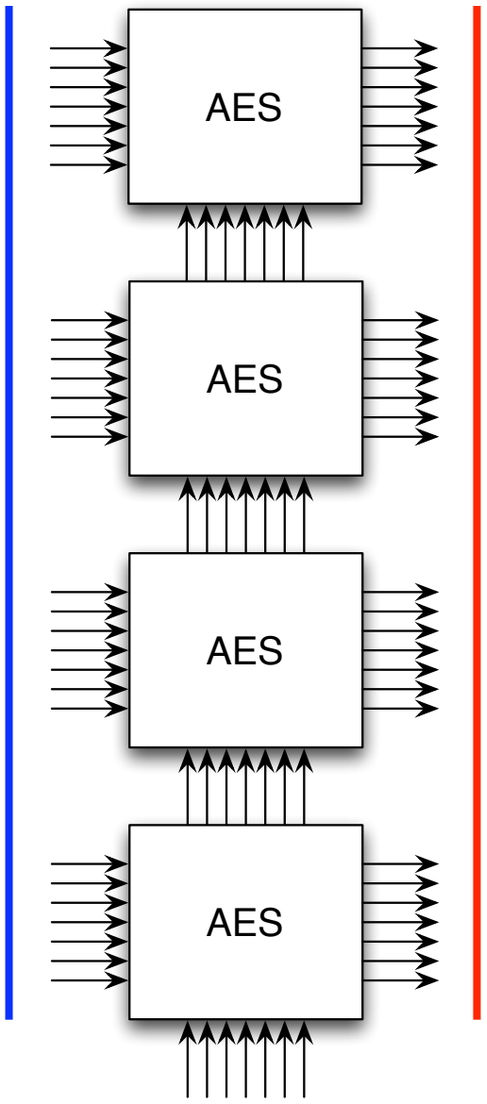


salattu sanoma (tunnettu)

avain (tuntematon)

Arvataan avain bitti kerrallaan – arvaus oikein jos piiri toteutuva

selväkielinen sanoma (tunnettu)



salattu sanoma (tunnettu)

avaimen alkuosa & arvausbitti (tunnettu)

avaimen loppuosa (tuntematon)

Kokonaisluvut

- **Kokonaislukujen tulo**

  Syöte: kokonaisluvut x ja y

  Tehtävä: määritä tulo xy

$$
\begin{array}{r}
1234 \\
\cdot\ 5678 \\
\hline
9872 \\
8638 \\
7404 \\
+\ 6170 \\
\hline
7006652
\end{array}
$$

- **Kokonaisluvun tekijöinti**

  Syöte: kokonaisluku x

  Tehtävä: määritä luvun x alkulukutekijät

$$7006652 = 2^2 \cdot 17 \cdot 167 \cdot 617$$

D. Bonenberger & M. Krone (29.12.2009)

RSA-170 =

260626236841398449215298792666744321970859253804864064161647851918599996285420693614502839319145146186835121981648059198820530572229741164780650958098323773365107115457593586420730428501486799804587268520423291459681059978161140231860633948450858040593963×726702906410701907886379776392394626413613780385699667031370893600228158224958749449 3

Vierailijan (ssh) RSA-modulus

```
24393870185186267499836722144675211065914
47169694413633277288167088933480094430052 1
24479943861233302261279501554881382342483 7
10019773446566337342152766455139374499318 0
95653425646769948724809696777544607874642 9
79593914382035272229882797444626741039608 8
52200988041875678106223647469458028927615 1
15294977574283152646377523808667376408605 4
88247951907562745734423031251507027715367 1
45879952757975425908200719084930340918408 8
24117918862694258943758338666993595924656 4
51551256481675938302156425982668819089855 6
37662394663490422273663204468558518785742 3
28158068854541964370707150150324371557505 8
76248597400449312510451541 49
```

617 desimaalia
(2048 bittiä)

solo1.nordea.fi RSA-modulus

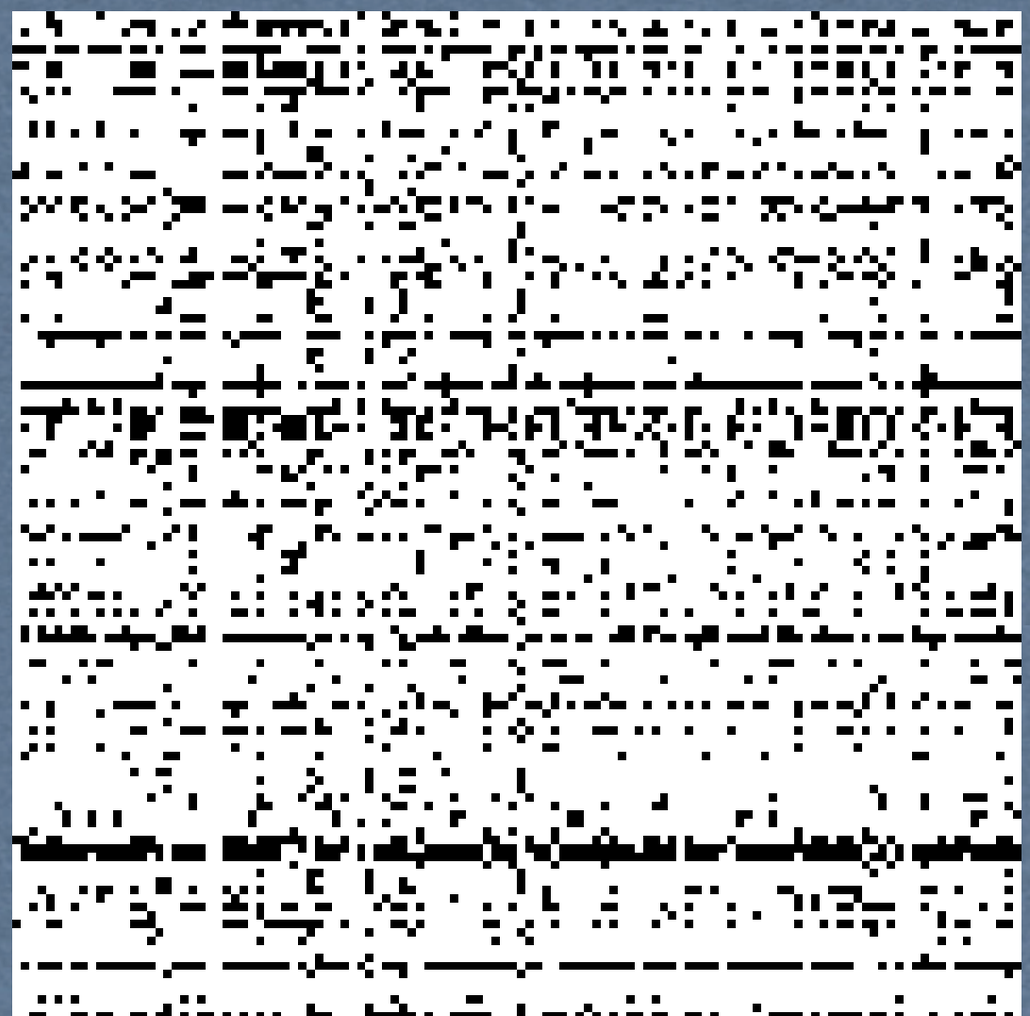
```
29146433424331574817373166388643222804400331404753109311876719602682484669805595624707995288605700693355876917529264503509386067037171378724024435163138623150575734370037461364077610719816220868172197035538339970494032314243980449463123881844661677487068167648605074407380915739214520114090734122736739832648448225771419026449230482187275373898623637533024742074278465518962955954417899713987192640101702469121254528156687728076867607752379064485580854461730137245078948178345598848507059445812824213167428222641917329086947683403453599875838574306753321725784747454472755605884118110763337137123576752373123133614931
```

617 desimaalia
(2048 bittiä)

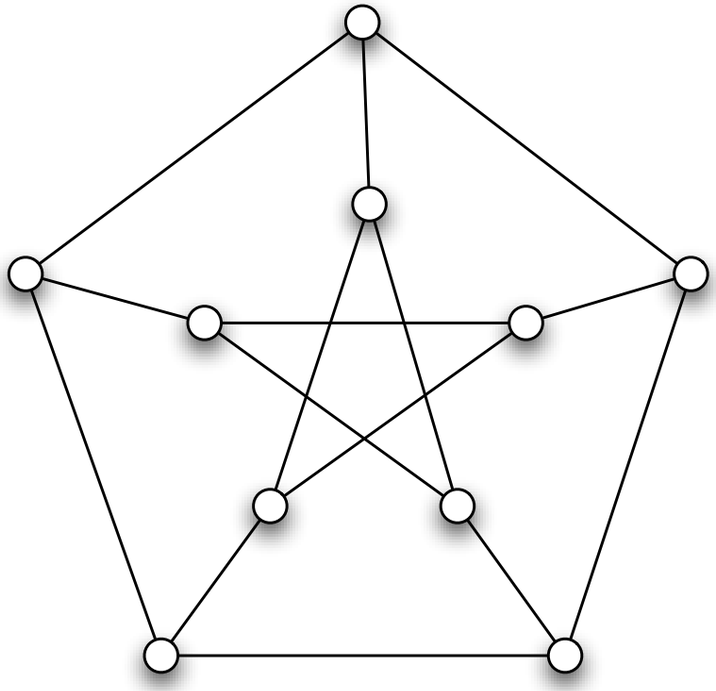
Matriisien yhtäsuuruus

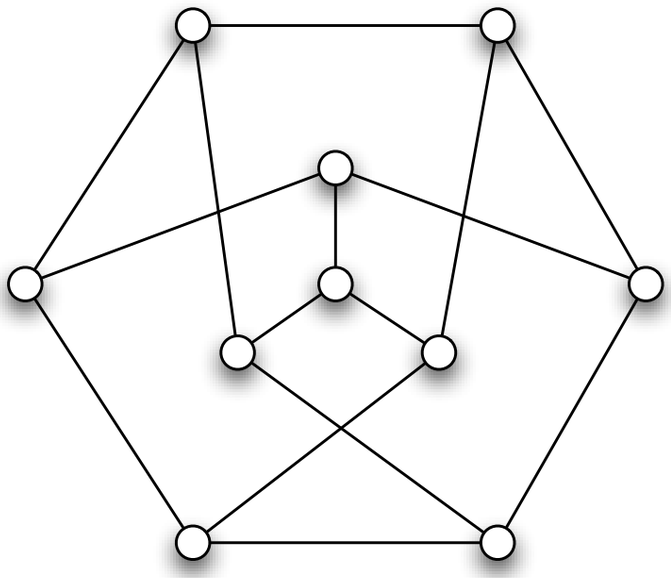
Sama matriisi?
(rivijärjestystä vaille)

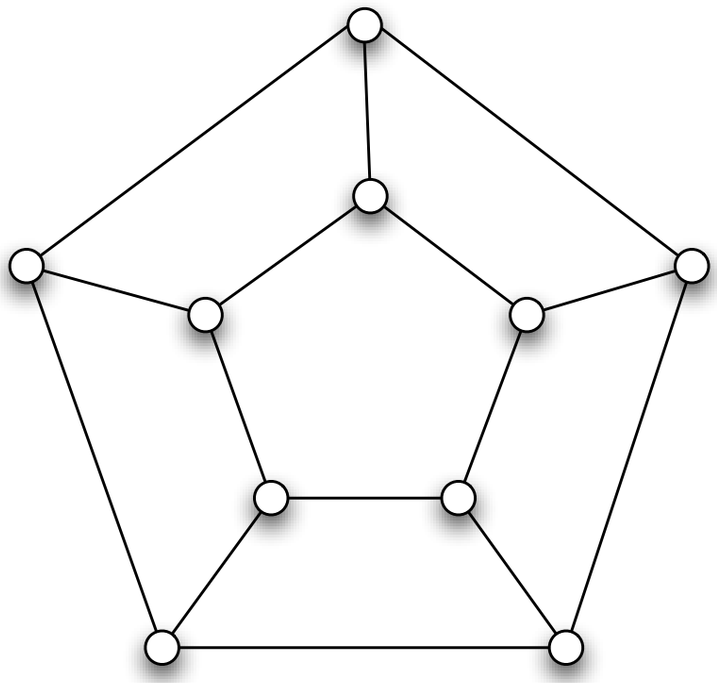
Sama matriisi?
(rivi- ja sarakejärj. vaille)



Esimerkki:
Onko kyseessä sama verkko?





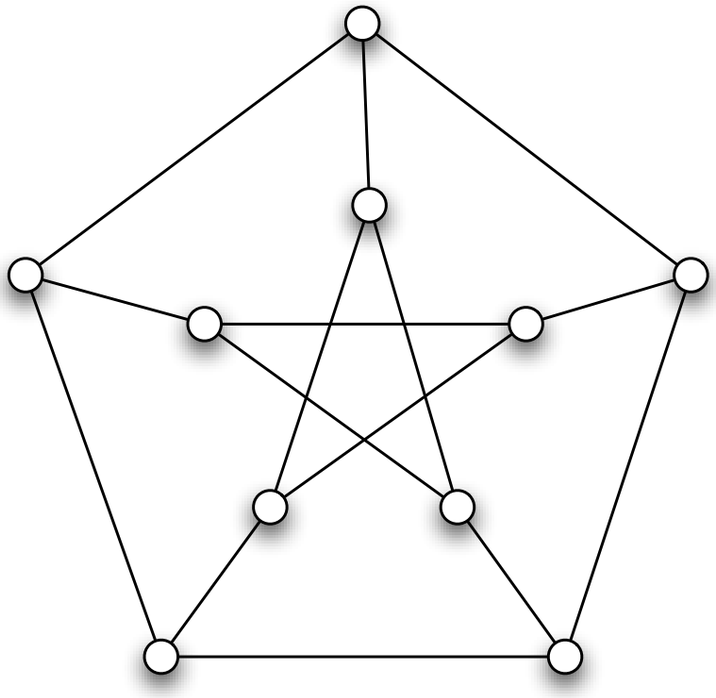


- Verkko-ongelmat
  - pienin virittävä puu
  - kauppamatkustajan ongelma

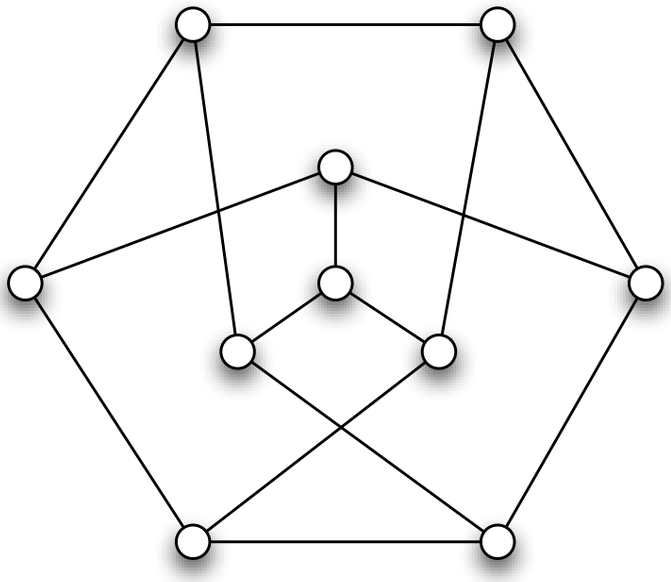
- Boolen logiikka
  - piirin arvo
  - piirin toteutuvuus

- Kokonaisluvut
  - kertolasku
  - tekijöinti

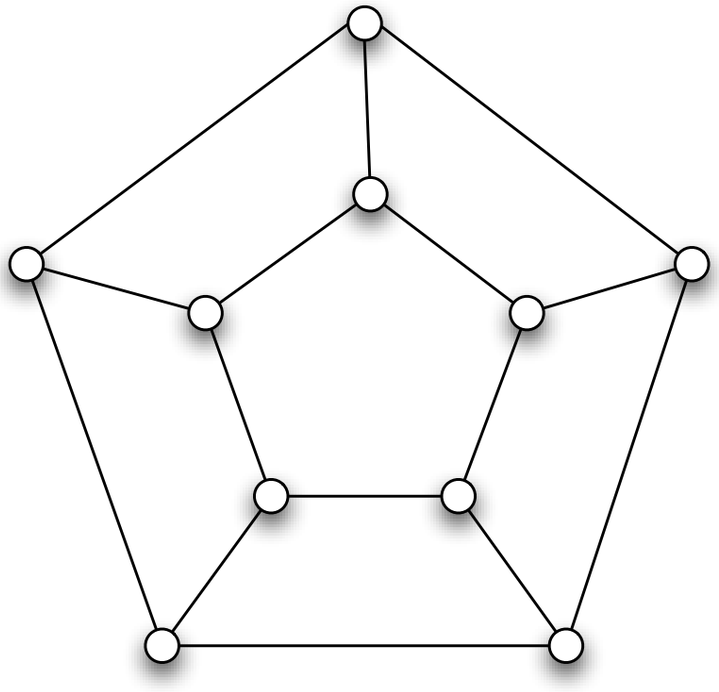
- Matriisien yhtäsuuruus
  - rivijärjestystä vaille
  - rivi- ja sarakejärjestystä vaille



A



B



C

Luennon sisältö

- Esimerkkejä laskennallisista ongelmista
- Laskennan vaativuus --- tehokas laskenta
- Ongelmaluokat P ja NP
- P=NP ?
- Luokan NP rakenteesta:
  NP-täydelliset ongelmat

2. Tehokas laskenta

- Kuinka tehokkaasti osaamme ratkaista ongelman?
- Tehokkuusmitta:
  Algoritmin pahimman tapauksen ajoaika
  syötteen koon (=n) funktiona

Esimerkkejä kokomitoista syötteelle:

- Verkko-ongelmat
  - pienin virittävä puu
  - kauppamatkustajan ongelma

  solmujen lkm

- Boolen logiikka
  - piirin arvo
  - piirin toteutuvuus

  porttien lkm

- Kokonaisluvut
  - kertolasku
  - tekijöinti

  lukujen pituus bitteinä

- Matriisien yhtäsuuruus
  - rivijärjestystä vaille
  - rivi- ja sarakejärjestystä vaille

  rivien ja sarakkeiden lkm

Tehokas ratkeavuus

- Laskennallinen ongelma on **tehokkaasti ratkeava** jos sille löytyy algoritmi, jonka pahimman tapauksen ajoaika kasvaa *enintään polynomisesti* syötteen koon *n* suhteen

Polynominen kasvu

- Syötteen koko kaksinkertaistuu
  => ajoaika kasvaa vakiokertoimella

n	n^2	n^3
2	4	8
4	16	64
8	64	512
16	256	4096
32	1024	32768
64	4096	262144
128	16384	2097152
256	65536	16777216
512	262144	134217728
1024	1048576	1073741824

Ylipolynominen kasvu

n	$n^{\log_2 n}$	2^n	$n!$
2	2	4	2
4	16	16	24
8	512	256	40320
16	65536	65536	$2.092 \cdot 10^{13}$
32	$3.355 \cdot 10^{7}$	$4.295 \cdot 10^{9}$	$2.631 \cdot 10^{35}$
64	$6.872 \cdot 10^{10}$	$1.845 \cdot 10^{19}$	$1.269 \cdot 10^{89}$
128	$5.630 \cdot 10^{14}$	$3.403 \cdot 10^{38}$	$3.856 \cdot 10^{215}$
256	$1.845 \cdot 10^{19}$	$1.158 \cdot 10^{77}$	$8.578 \cdot 10^{506}$
512	$2.418 \cdot 10^{24}$	$1.341 \cdot 10^{154}$	$3.477 \cdot 10^{1166}$
1024	$1.268 \cdot 10^{30}$	$1.798 \cdot 10^{308}$	$5.419 \cdot 10^{2639}$

Havainto (1/3)

- “Tehokas ratkeavuus” on teoreettinen yläraja sille mitä voidaan käytännössä ratkaista syötteen koon kasvaessa

Havainto (2/3)

- Ongelman tehokas ratkeavuus pysyy ennallaan vaikka syötteen koon määritelmää (polynomisesti) muutettaisiin
- Esimerkkejä:
  1. matriisin rivien ja sarakkeiden määrä
     vs. matriisin alkioiden määrä
  2. verkon solmujen määrä
     vs. verkon solmujen ja kaarien määrä
  3. kokonaisluvun pituus 10-järjestelmässä
     vs. binäärijärjestelmässä

Havainto (3/3)

- Tehokas ratkeavuus pysyy ennallaan vaikka laskennan mallia (jonka suhteen algoritmin ajoaikaa teoreettisesti mitataan) hieman muutettaisiin
- Esimerkki:

  Java-virtuaalikonetta voidaan simuloida Turingin koneella ja päinvastoin

- Yhden laskenta-askeleen simulointi vaatii simuloijalta syötteen kokoon nähden polynomisen määrän askelia

3. Ongelmien luokittelusta

Pienin virittävä puu

Piirin arvo

Kertolasku

Matriisien yhtäsuuruus (rivijärj. vaille)

Matriisien yhtäsuuruus (rivi- ja sarakejärj. vaille)

Kauppamatkustajan ongelma

Piirin toteutuvuus

Tekijöinti

Ongelmien luokittelusta

- Laskennallisia ongelmia voidaan luokitella monin eri kriteerein
- Ehkäpä keskeisin kriteeri on **tehokas ratkeavuus**
- Tavoitteenamme on jakaa laskennalliset ongelmat kolmeen luokkaan
  1. tehokkaasti ratkeaviin,
  2. ei-tehokkaasti ratkeaviin, ja

  (3. ratkeamattomiin)

- Verkko-ongelmat
  - pienin virittävä puu
  - kauppamatkustajan ongelma
- Boolen logiikka
  - piirin arvo
  - piirin toteutuvuus
- Kokonaisluvut
  - kertolasku
  - tekijöinti
- Matriisien yhtäsuuruus
  - rivijärjestystä vaille
  - rivi- ja sarakejärjestystä vaille

Tehokas ratkaisualgoritmi tunnetaan

Tehokasta algoritmia *ei tunneta*, *eikä sellaisen olemassaoloa ole pystytty poissulkemaan*

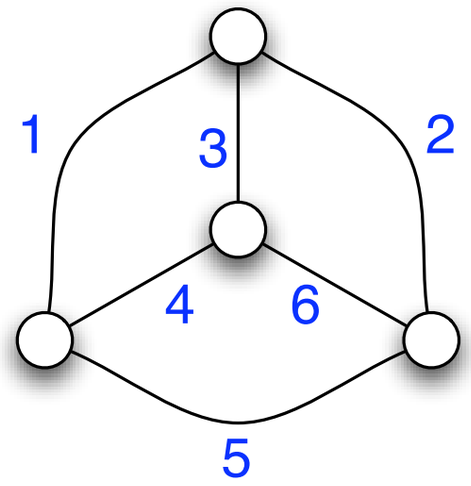
Ongelmaluokat

P ja NP

- Tarkastelemme jatkossa vain **päätösongelmia** (tehtävän ratkaisu: kyllä / ei)
- Esimerkki:

  Kauppamatkustajan ongelma (päätös)

  - Syöte:
    a) Kaaripainotettu täydellinen verkko
    b) Painoraja



- Tehtävä:
  Onko verkolla virittävää kehää, jonka kokonaispaino on enintään annettu painoraja? (kyllä / ei)

Luokka P

- Luokka **P** koostuu tehokkaasti ratkeavista päätösongelmista
- Esimerkkejä:
  - Pienin virittävä puu (päätös)
  - Piirin arvo on tosi?
  - Alkuluku?
  - Matriisien yhtäsuuruus rivijärjestystystä vaille

- Monille päätösongelmille ei tunneta tehokasta ratkaisualgoritmia, toisaalta sellaisen olemassaoloa ei ole pystytty poissulkemaan
- Esimerkkejä:
  - Kauppamatkustajan ongelma (päätös)
  - Piirin toteutuvuus
  - Matriisien yhtäsuuruus rivi- ja sarakejärjestystä vaille

- Onko seuraavilla päätösongelmilla yhteisiä piirteitä?
  - Kauppamatkustajan ongelma (päätös)
  - Piirin toteutuvuus
  - Matriisien yhtäsuuruus
    rivi- ja sarakejärjestystä vaille

="varmenne"

- Eräs yhteinen piirre:

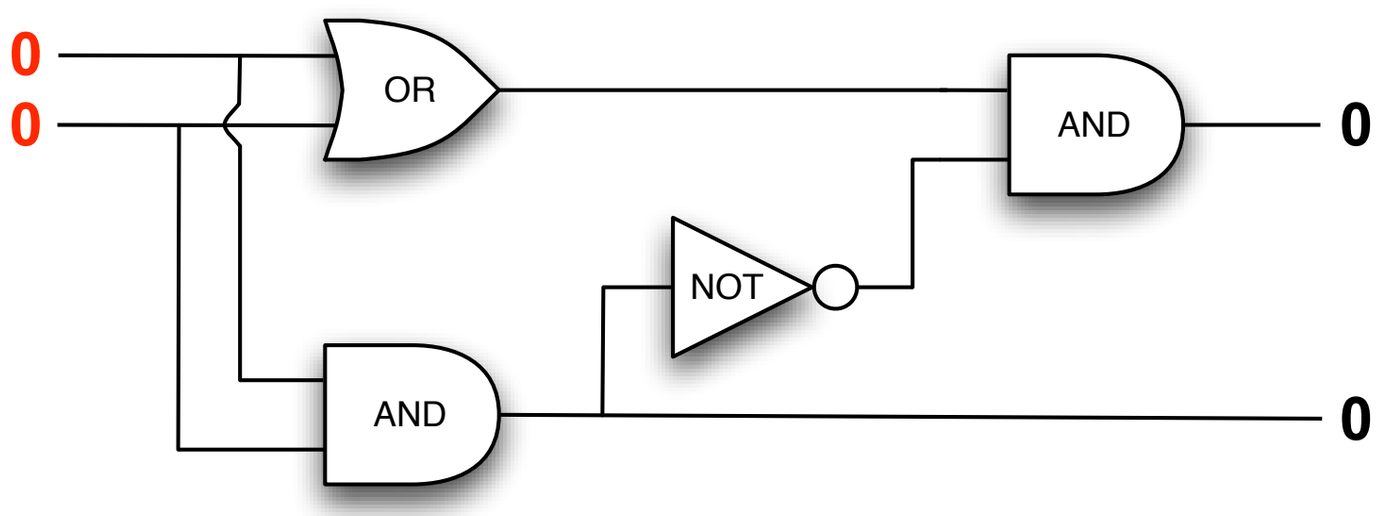
  Jos meille paljastetaan *sopivaa lisätietoa* annetusta *"kyllä"*-syötteestä, voimme *helposti varmistua* kyseessä olevan "kyllä"-syöte

Piirin toteutuvuus (päätös)

- Syöte:
  Piiri ja arvot piirin ulostuloille
  (sekä mahdollisesti osalle sisääntuloista)

- Tehtävä:
  Voidaanko tuntemattomille sisääntuloille asettaa arvot siten, että piiri saa halutut arvot? (kyllä/ei)

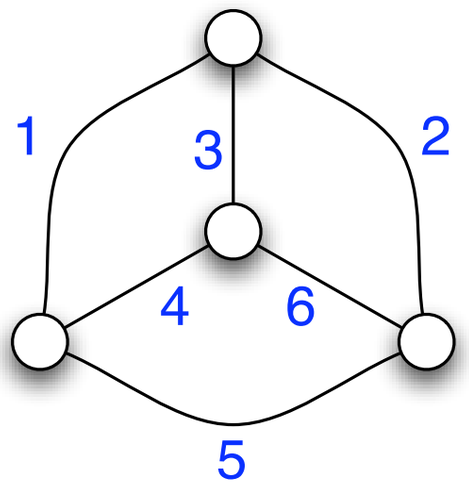
Varmenne
= arvot
tuntemattomille
sisääntuloille
s.e. piiri saa
halutut arvot

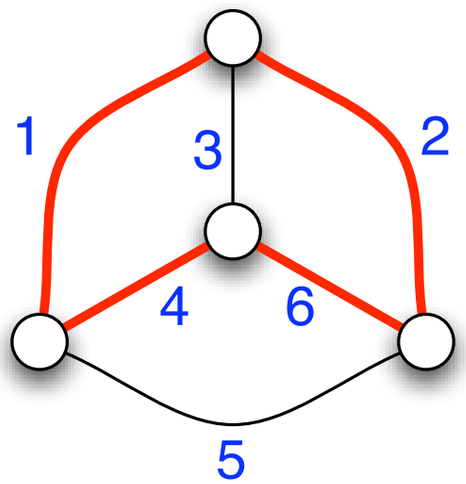


Kauppamatkustajan ongelma (päätös)

- Syöte:
  a) Kaaripainotettu täydellinen verkko
  b) Painoraja

- Tehtävä:
  Onko verkolla virittävää kehää, jonka kokonaispaino on enintään annettu painoraja? (kyllä / ei)

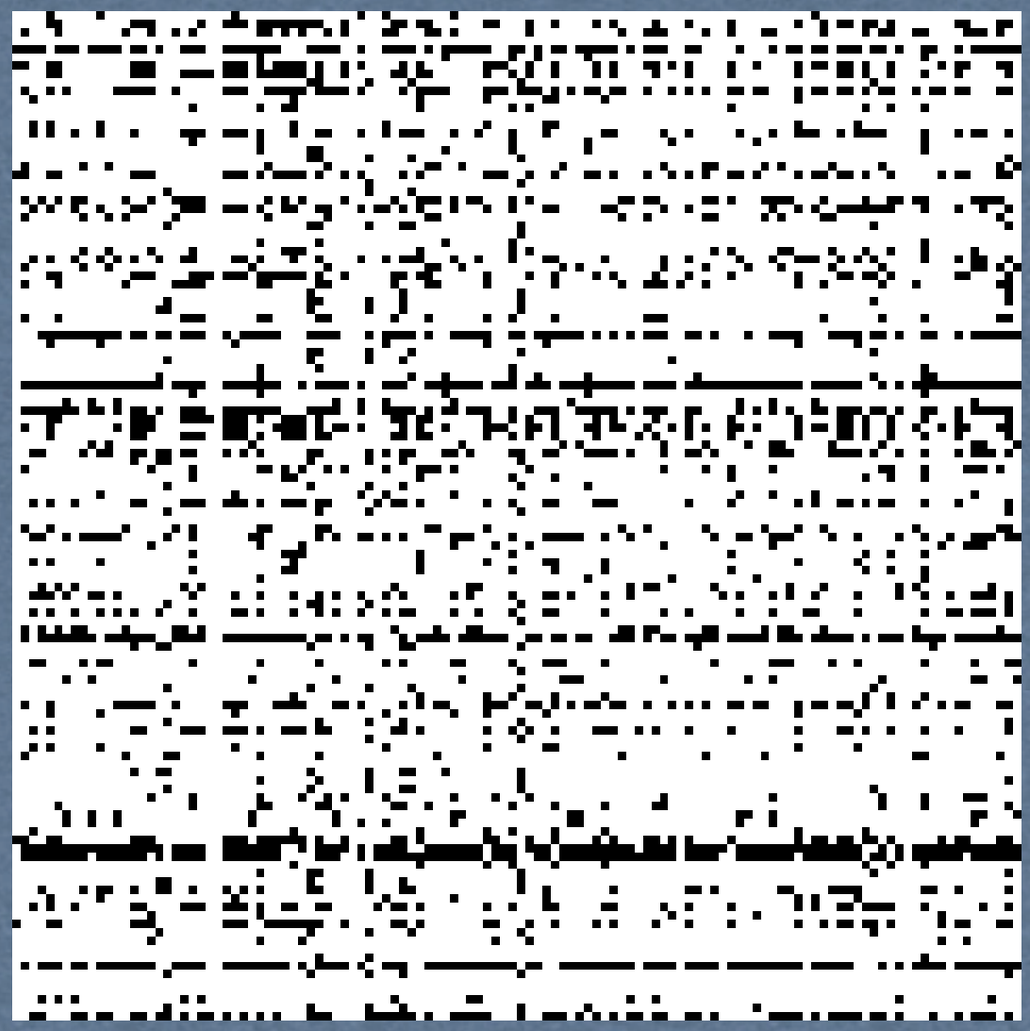




Varmenne
= painorajan toteuttava virittävä kehä

Matriisien yhtäsuuruus rivi- ja sarakejärjestystä vaille (päätös)

Varmenne = rivi- ja sarakepermutaatio,
joka muuntaa ensimmäisen matriisin toiseksi



- Päätösongelma on **tehokkaasti tarkistettavissa** jos on olemassa algoritmi, joka ottaa syötteekseen

  1) päätösongelman syötteen; ja
  2) *ehdokkaan* tämän varmenteeksi

  Algoritmilta vaaditaan, että

  a) päätösongelman "kyllä"-syötteen tapauksessa *on olemassa* ainakin yksi varmenne, joka tuottaa algoritmilta päätöksen "kyllä";

  b) päätösongelman "ei"-syötteen tapauksessa *kaikki* varmenne-ehdokkaat tuottavat algoritmilta päätöksen "ei";

  c) algoritmin ajoaika on enintään polynominen *päätösongelman* syötteen koon suhteen

Ratkeavuus *vs.* tarkistettavuus

- Jokainen tehokkaasti ratkeava päätösongelma on tehokkaasti tarkistettavissa
  - Perustelu: Tarkistusalgoritmiksi voidaan ottaa päätösongelman ratkaisualgoritmi
- *Ei tiedetä* onko jokainen tehokkaasti tarkistettavissa oleva päätösongelma tehokkaasti ratkeava

Luokka NP

- Luokka **NP** koostuu tehokkaasti tarkistettavissa olevista päätösongelmista
- Esimerkkejä:
  - Kauppamatkustajan ongelma (päätös)
  - Piirin toteutuvuus
  - Matriisien yhtäsuuruus rivi- ja sarakejärjestystä vaille
- Luokka **P** sisältyy luokkaan **NP**

P $\stackrel{?}{=}$ NP

"Onko ratkaisun löytäminen aidosti
työläämpää kuin sen tarkistaminen?"

- http://www.claymath.org/millennium/P_vs_NP/

...	...	.1.
4..	...	...
.2.	...	...
...	.5.	4.7
..8	...	3..
..1	.9.	...
3..	4..	2..
.5.	1..	...
...	8.6	...

693	784	512
487	512	936
125	963	874
932	651	487
568	247	391
741	398	625
319	475	268
856	129	743
274	836	159

Luennon sisältö

- Esimerkkejä laskennallisista ongelmista
- Laskennan vaativuus --- tehokas laskenta
- Ongelmaluokat P ja NP
- P=NP ?
- Luokan NP rakenteesta:
  NP-täydelliset ongelmat

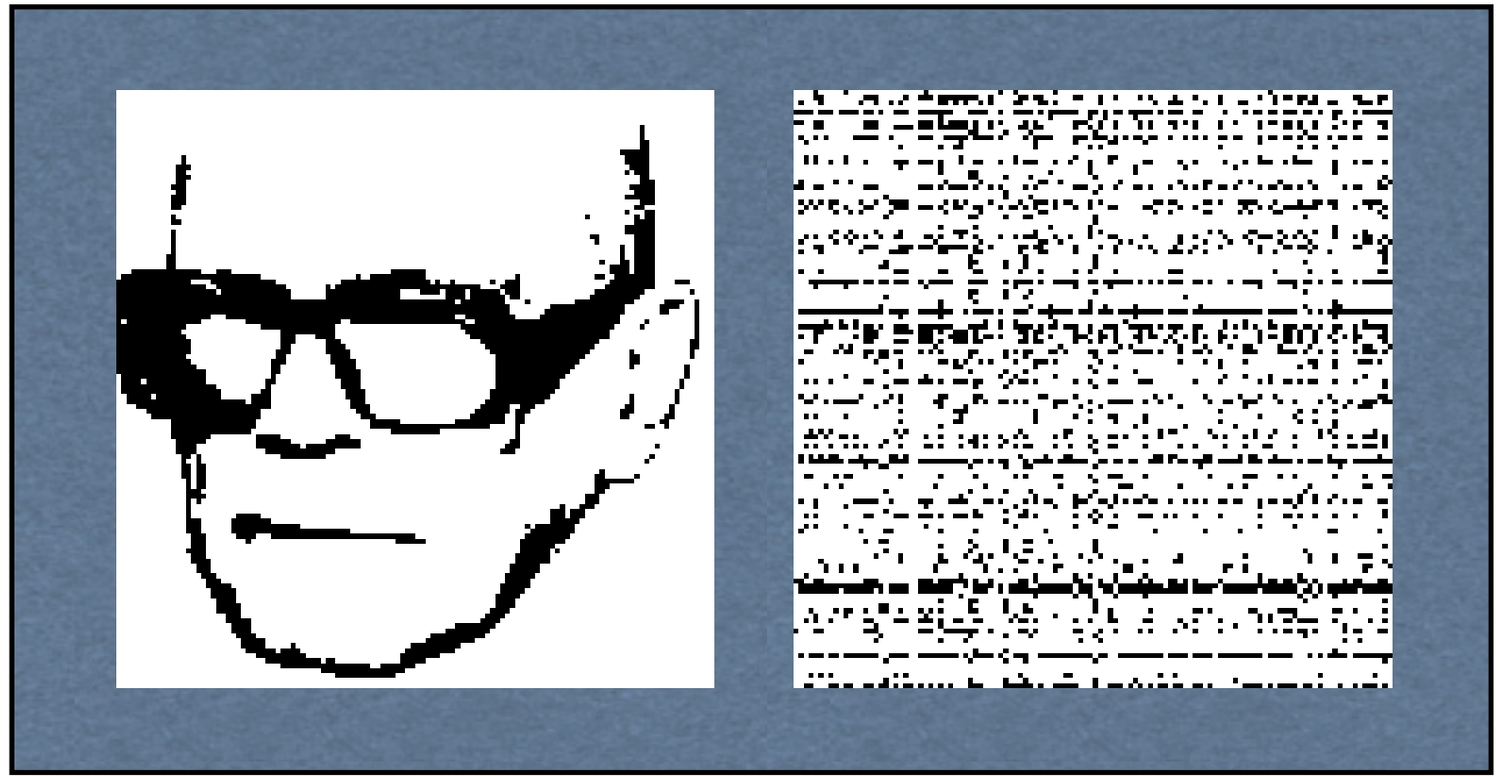
Ongelmien suhteellisesta työläydestä

- Onko päätösongelma A
  "enintään yhtä työläs kuin" päätösongelma B?
- Esimerkki:

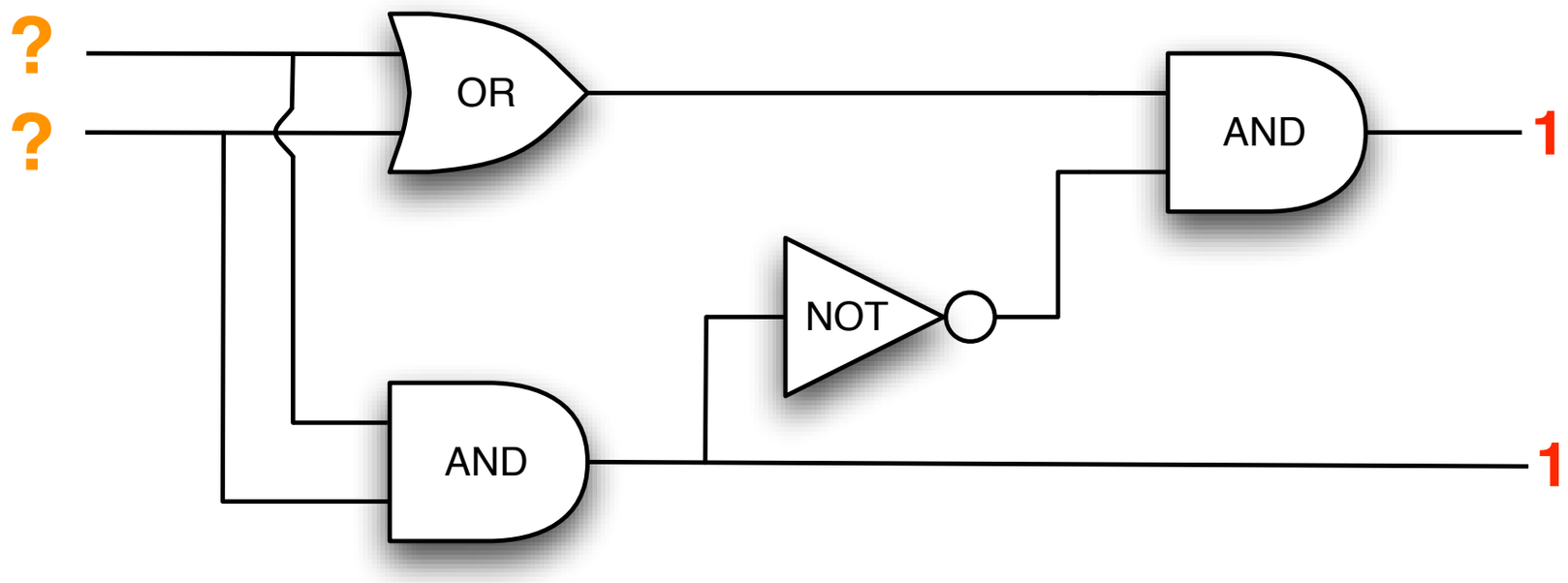
  A = Matriisien yhtäsuuruus (rivi- ja sar. järj. vaille)

  B = Piirin toteutuvuus
- Huom:
  - Molemmat luokassa NP
  - Kumpaankaan ei tunneta tehokasta algoritmia

Matriisien yhtäsuuruus (rivi- ja sar. järj. vaille)



Piirin toteutuvuus



Palautuksen käsite

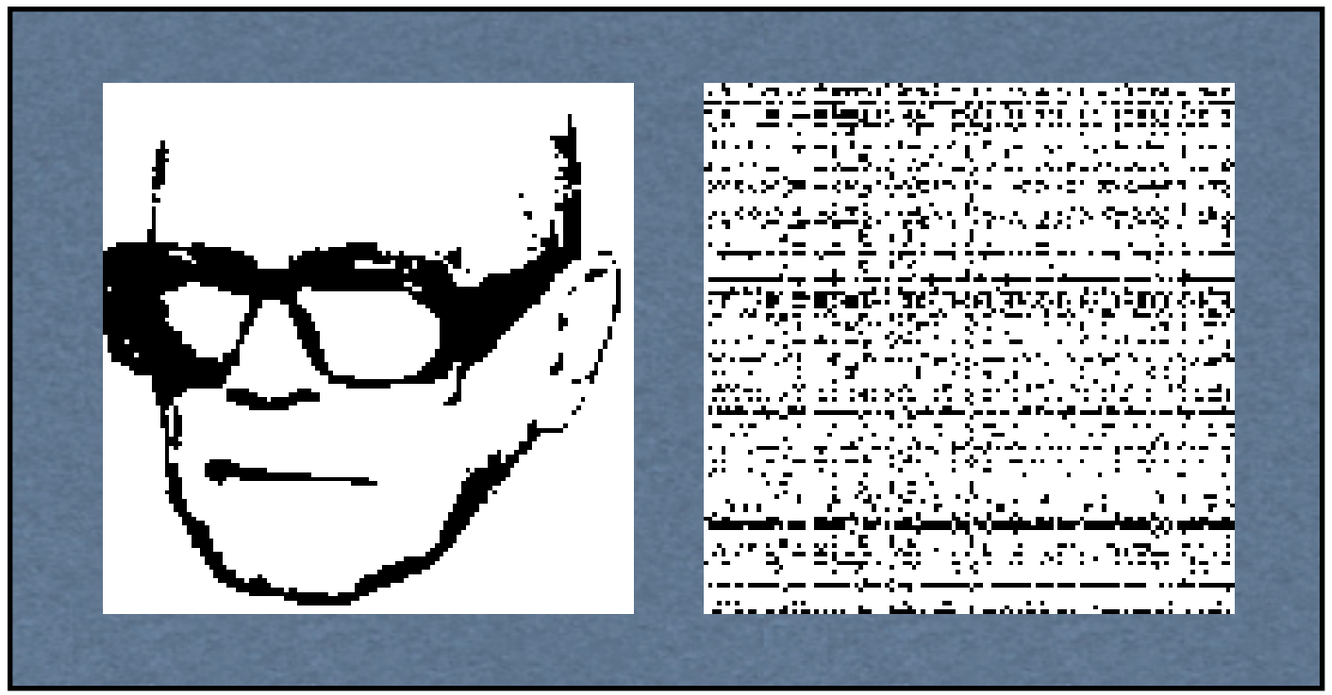
- Olkoon *P* algoritmi, joka ottaa syötteekseen päätösongelman A syötteen *x*, ja muodostaa tästä päätösongelman B syötteen *P*(*x*)
- Sanomme, että *P* on **palautus** päätösongelmasta A päätösongelmaan B jos pätee, että
  x on "kyllä"-syöte *jos ja vain jos*
  P(*x*) on "kyllä"-syöte
- Palautus *P* on **tehokas** jos sen ajoaika on enintään polynominen syötteen koon suhteen

- Oletetaan että päätösongelmasta A on olemassa tehokas palautus päätösongelmaan B
- Tällöin jos B on tehokkaasti ratkeava, niin on myös A
  - Perustelu:
    Muunnetaan annettu A:n syöte *x*
    B:n syötteeksi *P*(*x*) tehokkaalla palautuksella,
    ja ratkaistaan *P*(*x*) käyttäen B:n
    tehokasta ratkaisualgoritmia
- Tehokas palautus A ==> B
  ~ "A on enintään niin työläs kuin B"

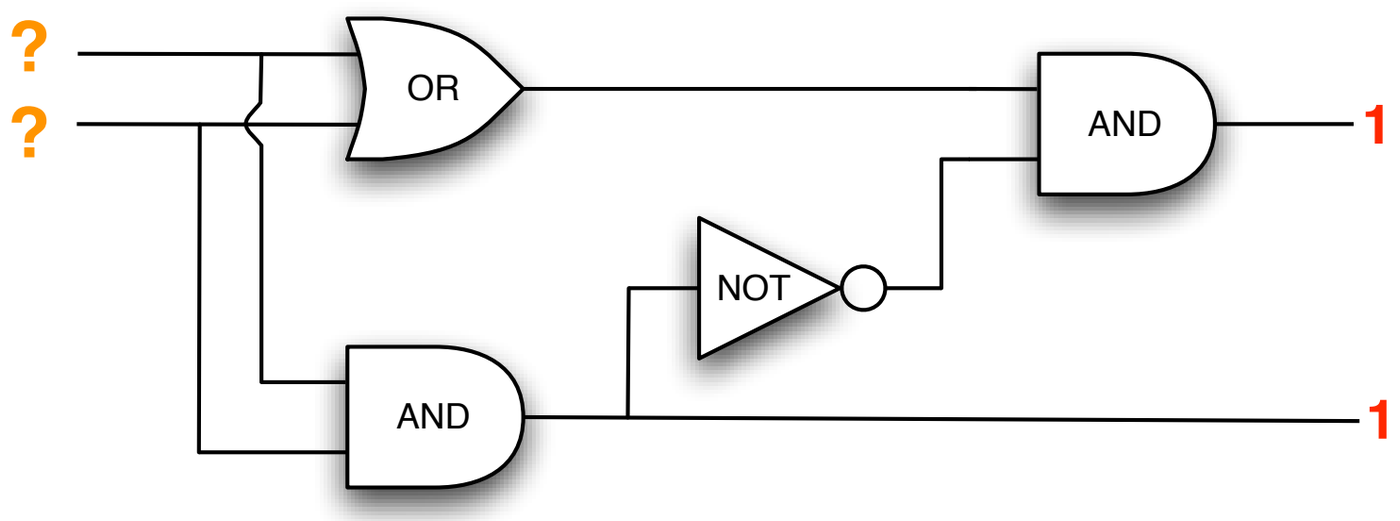
Tehokas palautus “Matriisien yhtäsuuruus” ==> “Piirin toteutuvuus”

Tehokas algoritmi *P*, joka muuttaa kaksi annettua matriisia piiriksi ja arvoiksi...

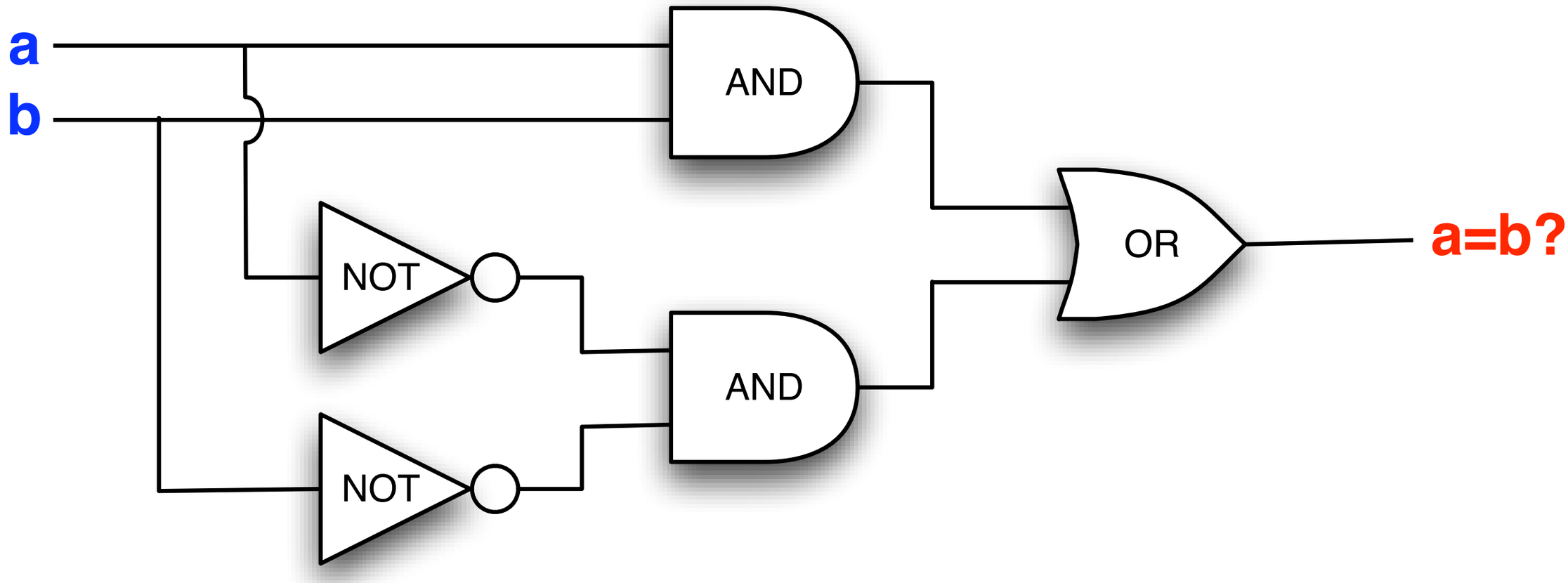
x:



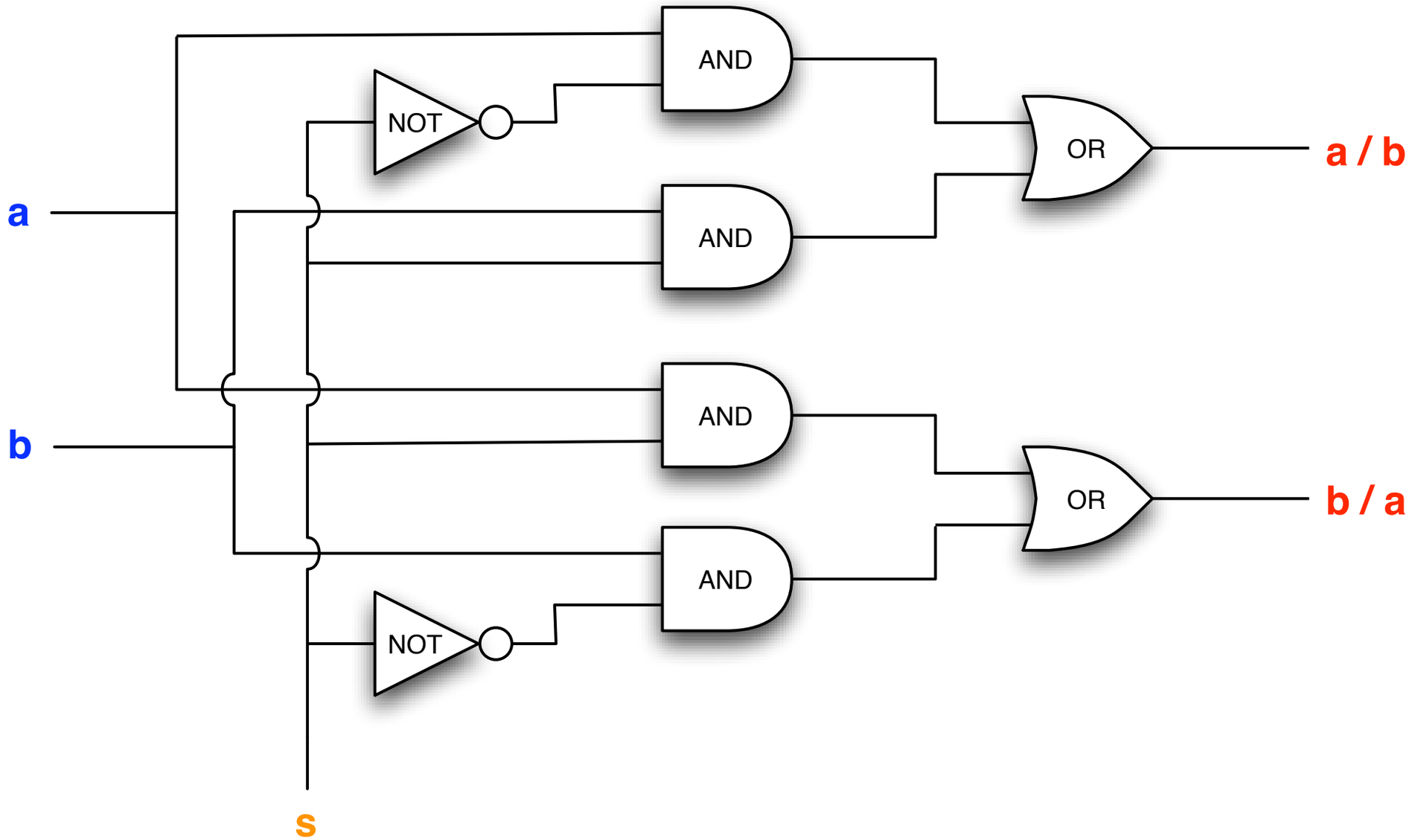
P(*x*):



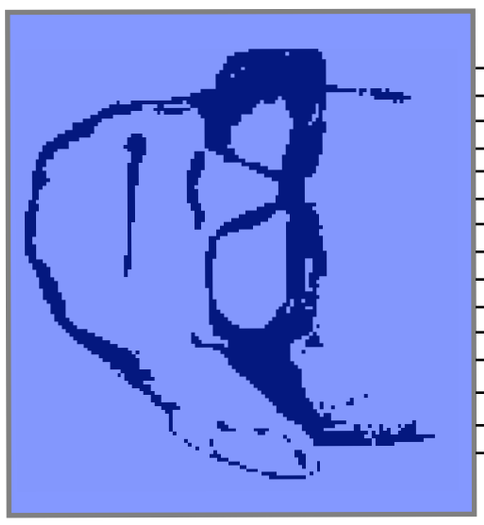
1) Piiri, joka tarkistaa yhtäsuuruuden

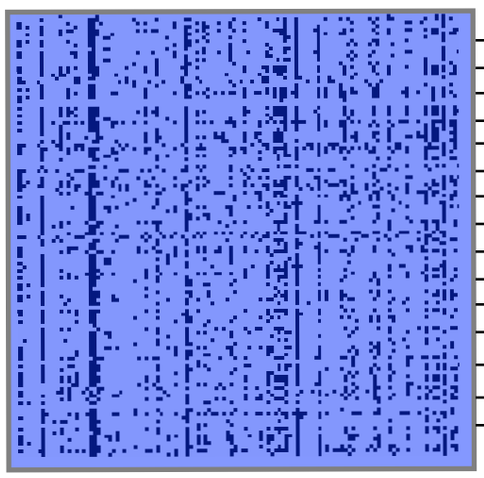


2) Piiri, joka *ohjatusti* vaihtaa kaksi syötettä keskenään



3) Piiri, joka on toteutuva *jos ja vain jos* matriisit ovat yhtäsuuria rivien ja sarakkeiden järjestystä vaille





Rivien ohj. vaihto pareittain

Sarakk. ohj. vaihto pareittain

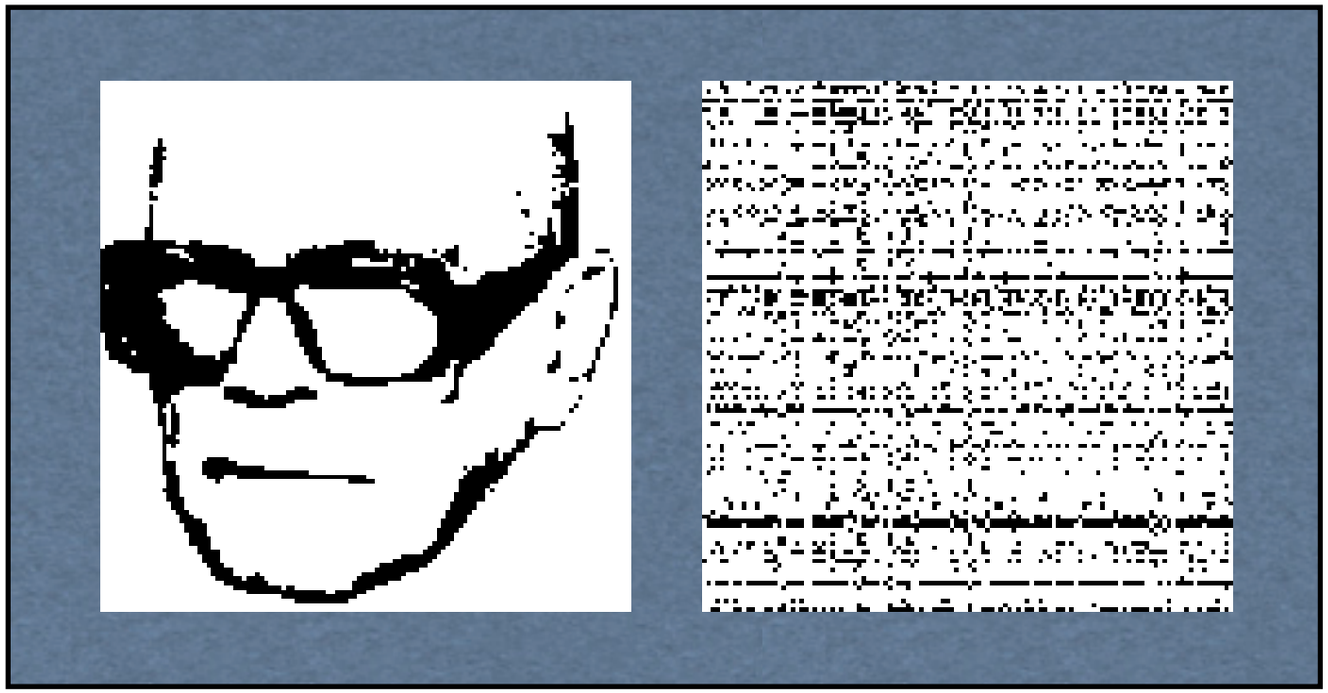
Yhtä-suuruus

1
1
1
1
1
1

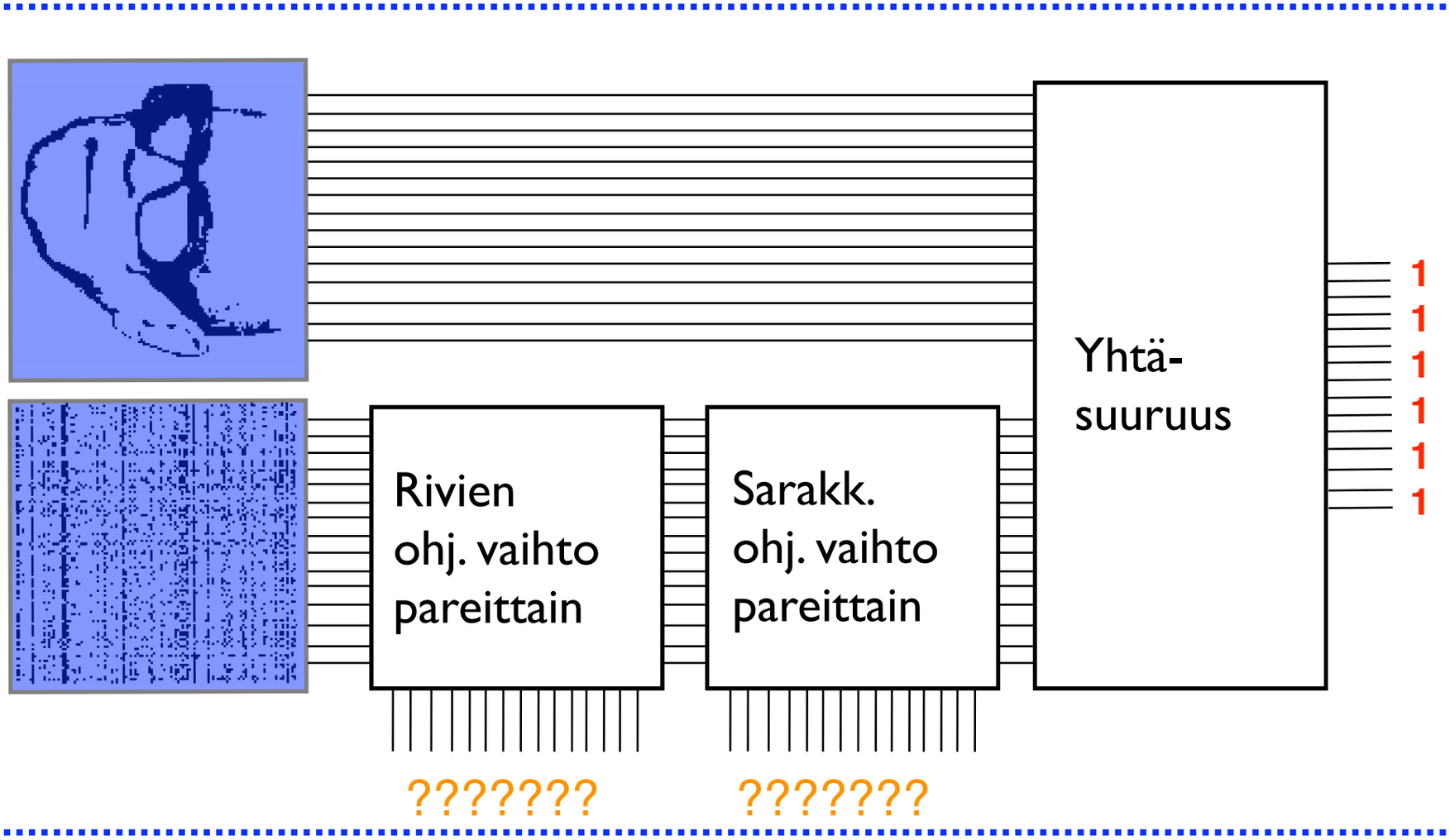
??????? ???????

Tehokas palautus

x:



P(*x*):



NP-täydelliset ongelmat

- Päätösongelma B on **NP-täydellinen** jos

  1) ongelma B on itse luokassa NP; ja

  2) *<u>jokaisesta</u>* luokan NP päätösongelmasta A
     on olemassa tehokas palautus ongelmaan B

- NP-täydelliset ongelmat ovat luokan NP "työläimpiä" ongelmia
- Erityisesti, jos jokin NP-täydellinen ongelma on tehokkaasti ratkeava, *kaikki* luokan NP ongelmat ovat tehokkaasti ratkeavia
- Perustelu:
  - Olkoon B NP-täydellinen ongelma, jolle on olemassa tehokas ratkaisualgoritmi
  - Olkoon A jokin päätösongelma luokassa NP
  - Koska B on NP-täydellinen, on olemassa tehokas palautus A ==> B
  - Voimme tehokkaasti ratkaista A:n palauttamalla ongelmaan B, ja käyttämällä B:n ratkaisualgoritmia

NP-täydelliset ongelmat

- NP-täydellisiä ongelmia on olemassa!

- **Lause** (Cook 1971, Levin 1973)
  Piirin toteutuvuus on NP-täydellinen

- Luokan NP päätösongelma A
  voidaan edellisen perusteella osoittaa
  NP-täydelliseksi esittämällä vain tehokas
  palautus B ==> A,
  missä B on NP-täydellinen

Kauppamatkustajan ongelma (päätös)

- Syöte:
  a) Kaaripainotettu täydellinen verkko
  b) Painoraja

- Tehtävä:
  Onko verkolla virittävää kehää, jonka kokonaispaino on enintään annettu painoraja? (kyllä / ei)

NP-täydellinen

Riippumaton joukko (päätös)

- Syöte:
  a) Verkko
  b) Kokonaisluku *k*
- Tehtävä:
  Löytyykö verkosta *k* solmun joukko,
  joiden välillä ei ole kaaria? (kyllä / ei)

NP-täydellinen

Solmupeite (päätös)

- Syöte:
  a) Verkko
  b) Kokonaisluku *k*

- Tehtävä:
  Löytyykö verkosta *k* solmun joukko siten, että jokaisen kaaren ainakin toinen päätepiste on joukossa?

NP-täydellinen

Verkon solmuväritys (päätös)

- Syöte:
  a) Verkko
  b) Kokonaisluku *k*

- Tehtävä:
  Voidaanko verkon solmut värittää *k* värillä siten, että jokaisen kaaren päätepisteillä on eri väri?

NP-täydellinen

Verkon kaariväritys (päätös)

- Syöte:
  a) Verkko
  b) Kokonaisluku *k*
- Tehtävä:
  Voidaanko verkon kaaret värittää *k* värillä siten, että jokaiseen solmuun liittyvillä kaarilla on eri värit?

NP-täydellinen

Ja niin edelleen...

- ... NP-täydellisiä ongelmia tunnetaan varovaisesti arvioiden ainakin joitakin tuhansia ...
- Valitettavan moni käytännön kannalta hyvinkin keskeinen laskennallinen ongelma on NP-täydellinen
- Mikä harmillisinta, emme osaa sanoa ovatko nämä ongelmat tehokkaasti ratkeavia vai eivät!
- Toisaalta, jos jokin NP-täydellinen ongelma olisi tehokkaasti ratkeava, kaikki luokan NP ongelmat olisivat tehokkaasti ratkeavia...

Luokan NP rakenteesta

- *Verkkoisomorfia* (~matriisien yhtäsuuruus rivien ja sarakkeiden järj. vaille)
  on eräitä harvoja luonnollisia esimerkkejä luokan NP ongelmasta

  a) jolle ei tunneta tehokasta algoritmia; ja

  b) jota ei ole osoitettu NP-täydelliseksi

- Jos P≠NP, voidaan osoittaa, että luokassa NP on (ainakin keinotekoisia) ongelmia, jotka eivät ole NP-täydellisiä eivätkä luokassa P

- Verkko-ongelmat
  - pienin virittävä puu
  - **kauppamatkustajan ongelma**
- Boolen logiikka
  - piirin arvo
  - **piirin toteutuvuus**
- Kokonaisluvut
  - kertolasku
  - tekijöinti
- Matriisien yhtäsuuruus
  - rivijärjestystä vaille
  - rivi- ja sarakejärjestystä vaille

Tehokas ratkaisualgoritmi tunnetaan

NP-täydellisiä (päätösongelmina)

Tehokasta algoritmia *ei tunneta*, *eikä sellaisen olemassaoloa ole pystytty poissulkemaan*

Lisää aiheesta

- T-79.1001
  Tietojenkäsittelyteorian perusteet
- T-79.5103
  Computational Complexity Theory

- M.R. Garey, D.S. Johnson, *Computers and Intractability*:
  A Guide to the Theory of NP-Completeness.
  W. H. Freeman and Co., 1979
- C.H. Papadimitriou, *Computational Complexity*.
  Addison-Wesley, 1994
- S. Arora, B. Barak, *Computational Complexity*:
  A Modern Approach. Cambridge University Press, 2009