

# OSNOVE UMETNE INTELIGENCE

2021/22

*strojno učenje*

*nenadzorovano učenje*

*preiskovanje*

# Pridobljeno znanje s prejšnjih predavanj

- **strojno učenje**
  - **nomogrami** kot orodje za vizualizacijo naivnega Bayesovega klasifikatorja:
    - razumevanje vizualizacije: predstavitev prispevkov vrednosti posameznih atributov
    - točke, izračunane z razmerjem verjetja
    - os za vsak atribut, os za vsoto vseh točk
  - **metoda k najbližjih sosedov**
    - leno učenje
    - merjenje razdalje (razdalja Minkowskega - evklidska, manhattanska, hammingova razdalja)
    - izbira ustreznega k (prešibko ali pretirano prilagajanje)
    - normalizacija, prekletstvo dimenzionalnosti
  - **lokalno utežena regresija**
    - uteževanje označb z razdaljo
    - uporaba jedra, nastavitev širine jedra
  - **regresijska drevesa**
    - srednja kvadratna napaka (MSE) kot mera nečistoče

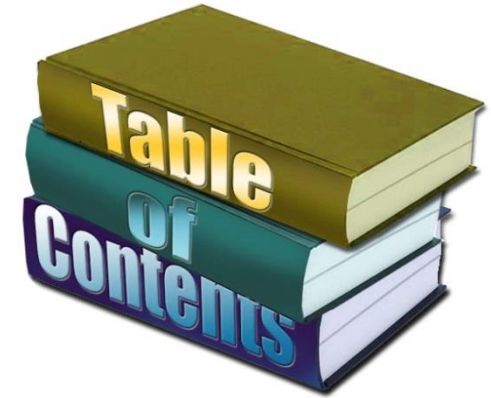
# Pregled

## I. strojno učenje

- uvod v strojno učenje
- učenje odločitvenih dreves
- učenje dreves iz šumnih podatkov (rezanje dreves)
- ocenjevanje učenja
- diskretizacija atributov, obravnava manjkajočih vrednosti
- naivni Bayesov klasifikator
- nomogrami
- k najbližjih sosedov
- lokalna utežena regresija
- regresijska drevesa
- linearni modeli
- hierarhično gručenje
- gručenje z metodo k voditeljev

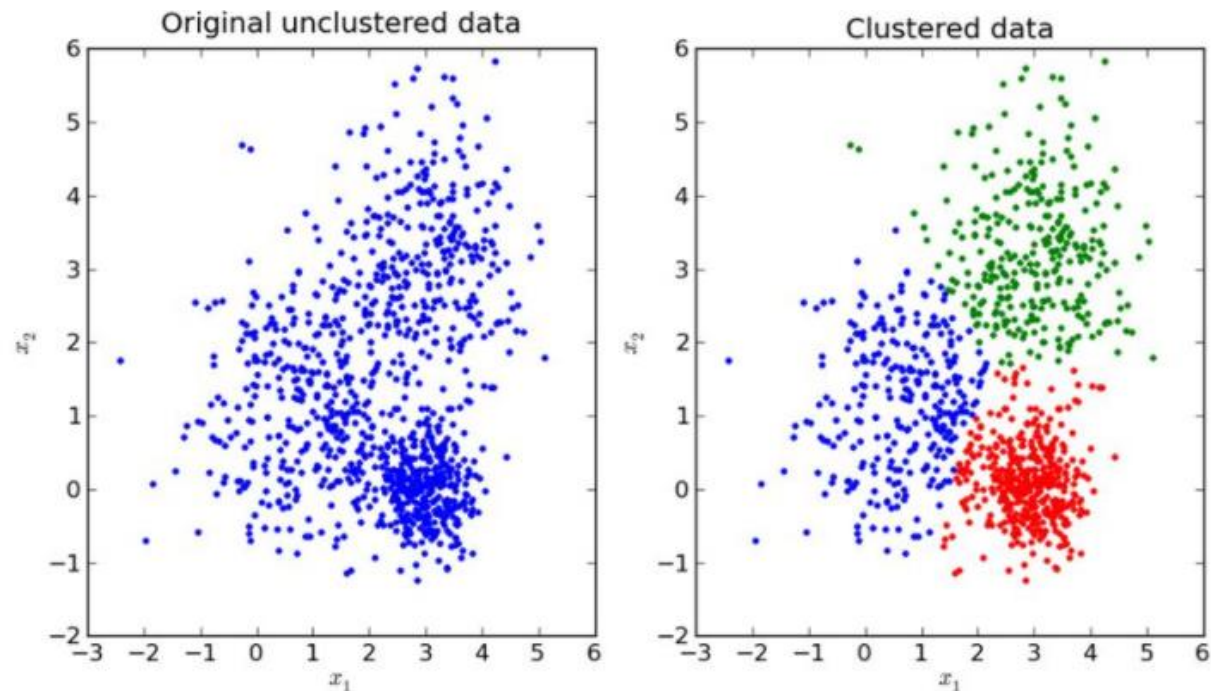
## II. preiskovanje

- definicija in cilji področja
- primeri umetnih in realnih problemov

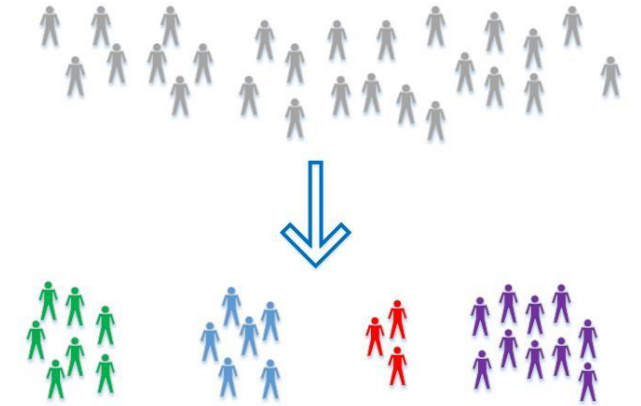


# Nenadzorovano učenje

- drugačni scenarij in cilji učenja kot pri nadzorovanem učenju:
  - **nimamo ciljne (odvisne) spremenljivke**, zato nas ne zanima napoved primera
  - podani so samo atributi primerov
- cilj: odkrivanje zakonitosti glede porazdelitve učnih primerov. Vprašanja:
  - ali lahko primere razdelimo v **smiselne skupine**?
  - ali obstaja priročen način za **vizualizacijo** podatkov?



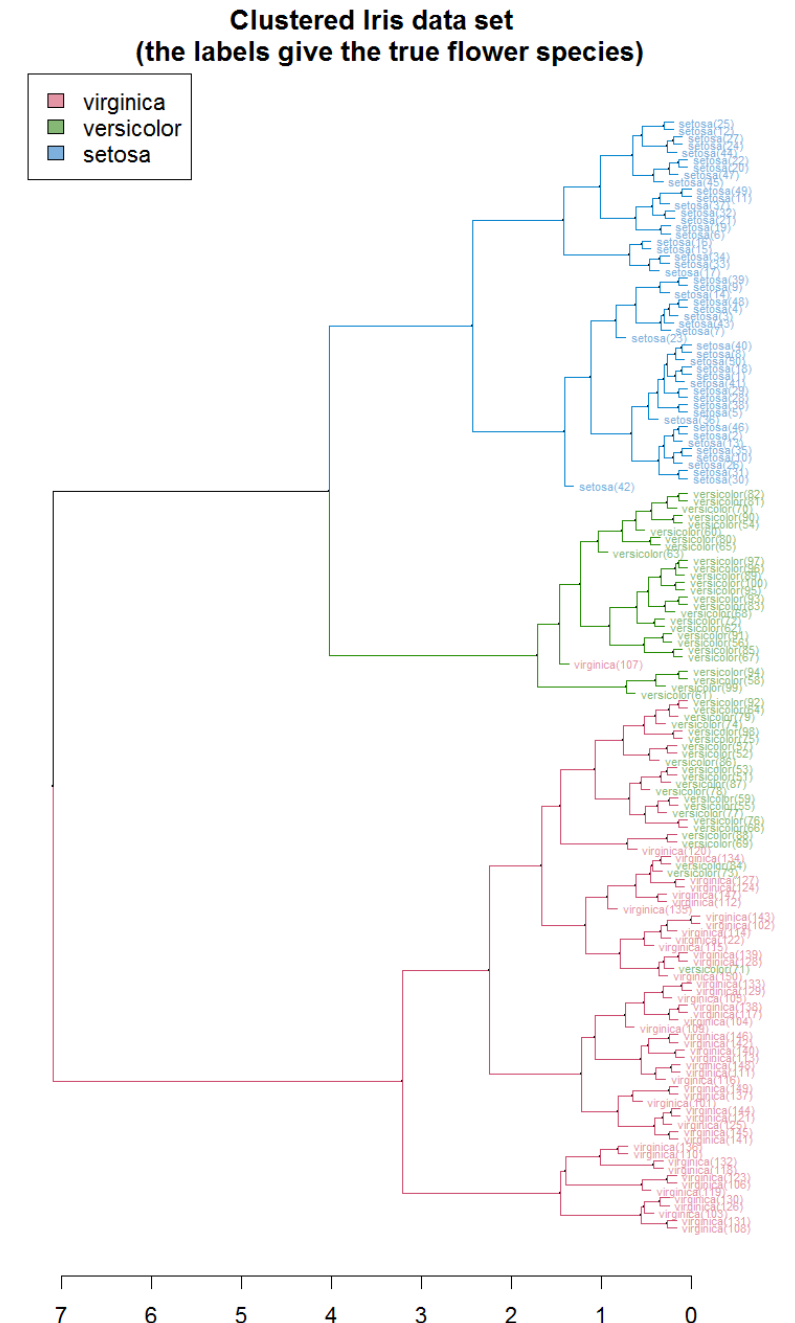
# Nenadzorovano učenje



- **lastnosti:**
  - nenadzorovano učenje je **bolj subjektivno** kot nadzorovano učenje, ker nima enoznačnega formalnega cilja kot je "napovedovanje vrednosti odvisne spremenljivke" pri nadzorovanem učenju
  - velikokrat lažje (**cenejše**) **pridobimo neoznačene podatke** (podatke brez odvisne spremenljivke): drage meritve, ekspertno mnenje, globalna ocena (npr. filma)?
- **primeri uporabe:**
  - odkrivanje skupin rakavih bolnikov, grupiranih po različnih rezultatih meritev izraženosti genov,
  - odkrivanje skupin kupcev, grupiranih po njihovi zgodovini brskanja in nakupovanja
  - odkrivanje skupin filmov, grupiranih glede na ocene, podane s strani gledalcev

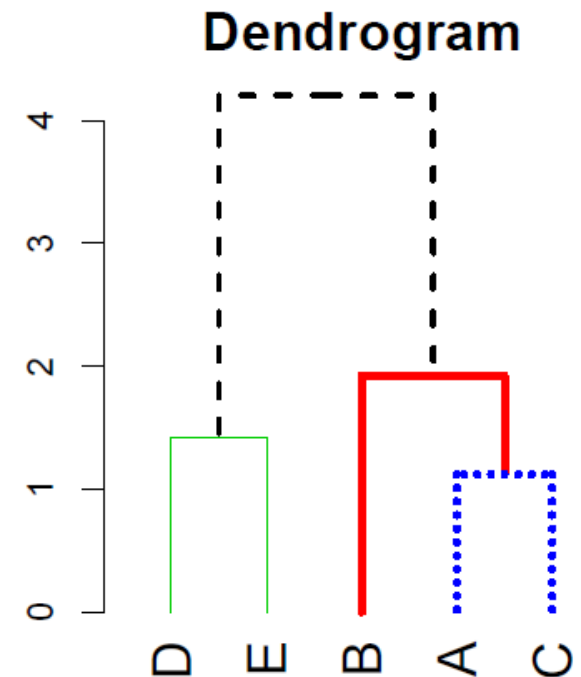
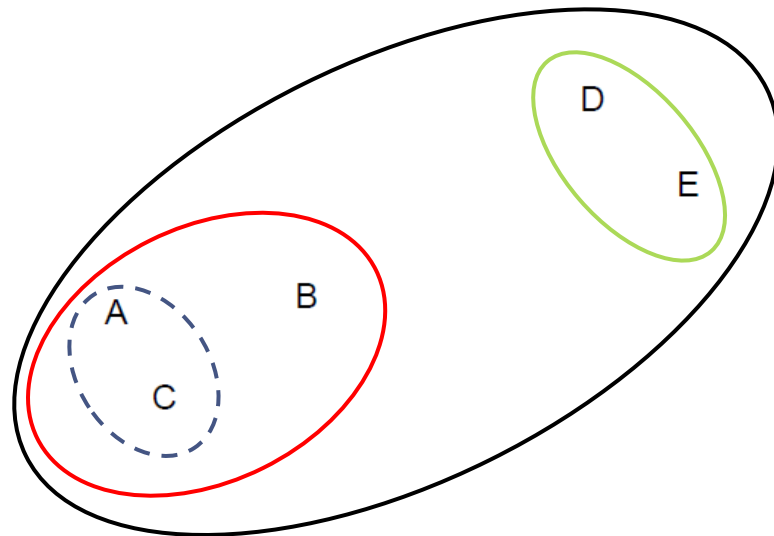
# Gručenje

- **gručenje (angl. clustering)** je najbolj uporabljana metoda nenadzorovanega učenja
- **cilj:** iskanje homogenih podskupin v učnih podatkih
- metode:
  - **hierarhično gručenje:** iščemo vnaprej neznanu število gruč. Rezultat gručenja je vizualna reprezentacija skupin, imenovana dendrogram, ki nam nudi vpogled v oblikovanje različnega števila gruč
  - **metoda k-means:** optimizacijski algoritem, ki poskuša iterativno primere gručiti v vnaprej podano število  $k$  gruč



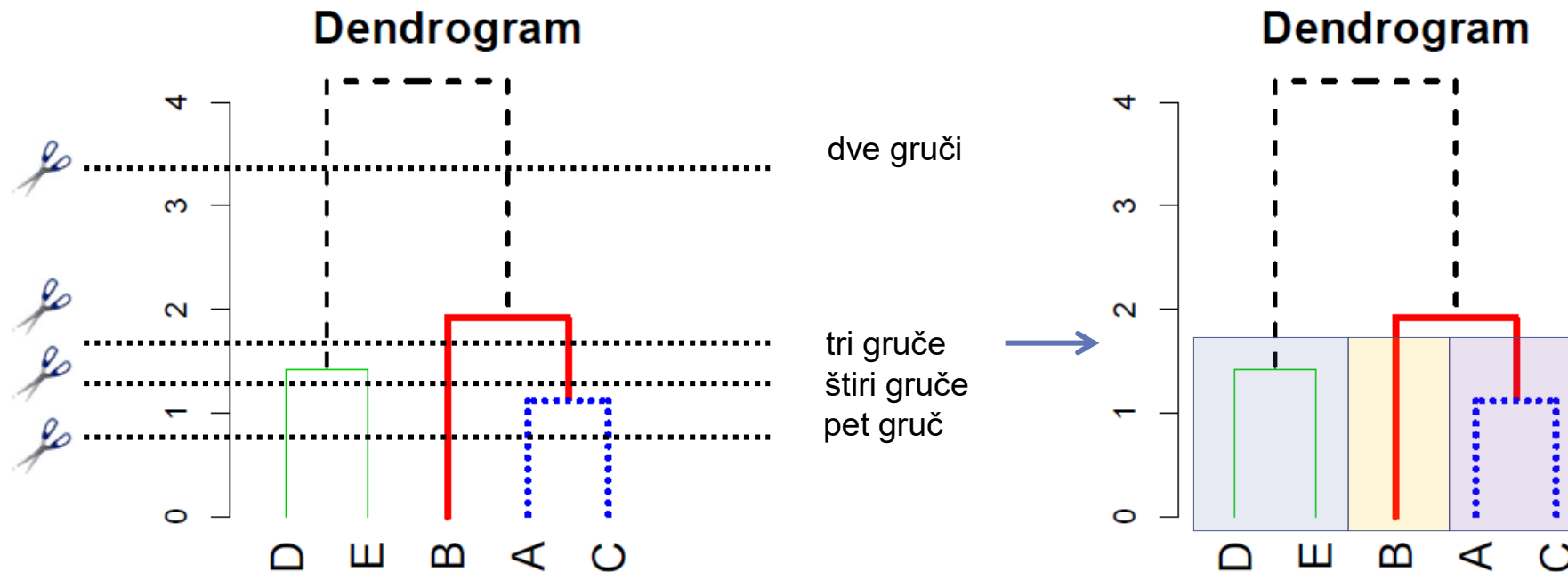
# Hierarhično gručenje

- dva pristopa:
  - **združevalni** (angl. *agglomerative*): gradnja dendrograma začnši od listov proti korenu s postopkom **združevanja** glede na razdaljo
  - **delilni** (angl. *divisive*): gradnja dendrograma od korena proti listom, na vsakem koraku **delimo** gručo na podgruče
- primer združevalnega pristopa:
  1. začni z vsako točko v svoji gruči
  2. najdi dve najbližji gruči in ju združi
  3. ponavljaj, dokler ne združiš vseh gruč



# Interpretacija dendrograma

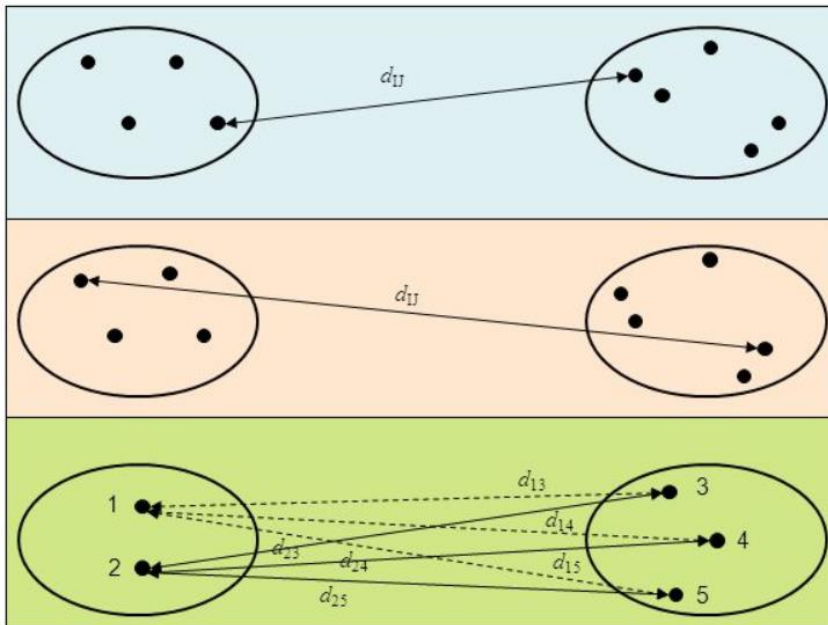
- rezanje dendrograma določi mejo, pri kateri prenehamo z združevanjem gruč
- z rezanjem dendrograma na različnih višinah torej določamo število ciljnih gruč





# Merjenje razdalj

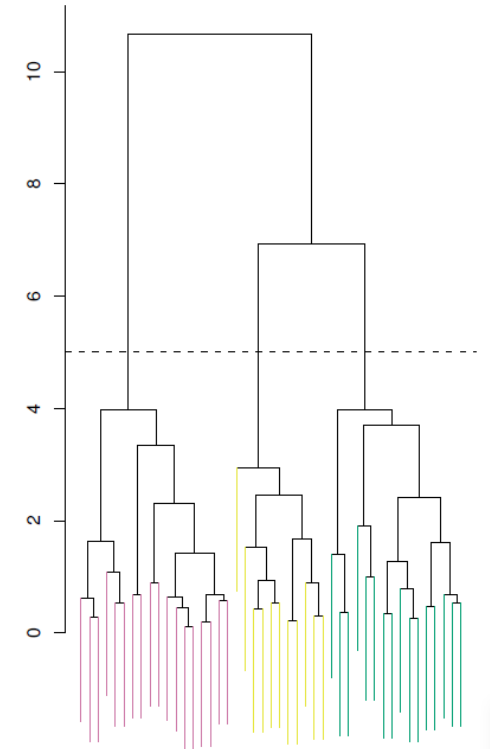
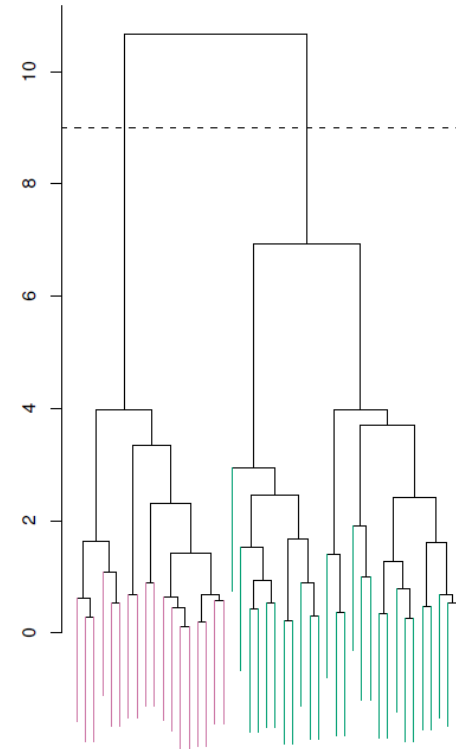
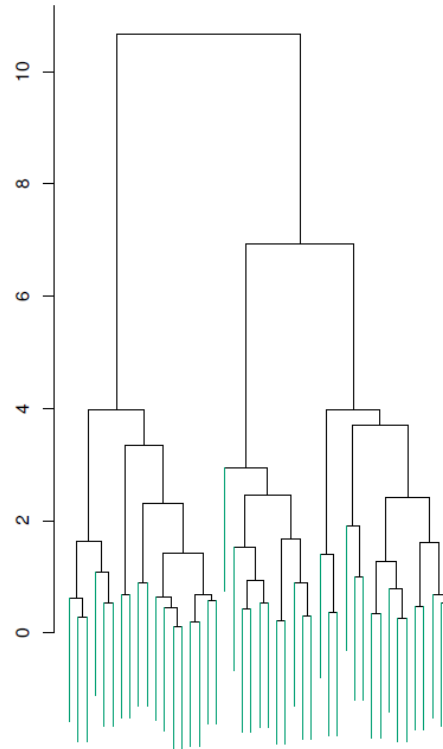
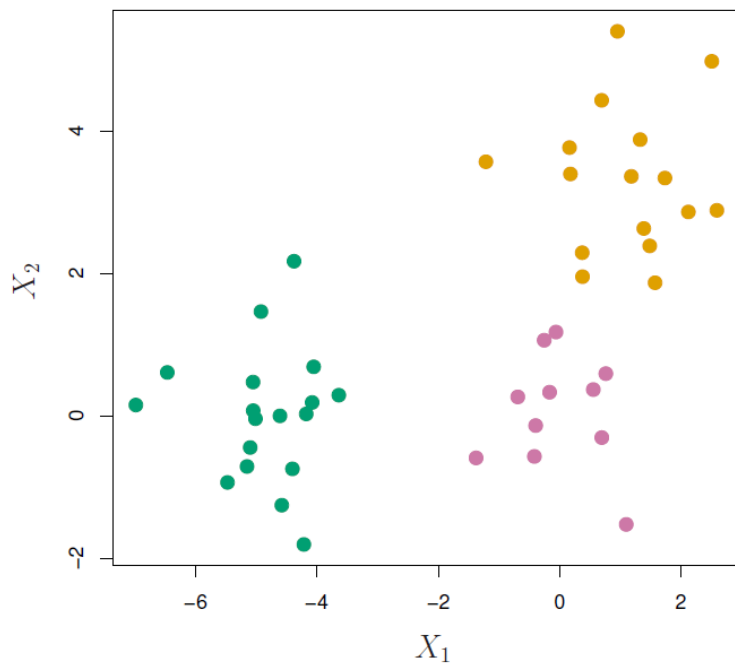
- med učnimi primeri uporabljamo že znane mere za merjenje razdalj (evklidska razdalja, manhattanska razdalja, korelacija med vrednostmi atributov ...)
- posebno obravnavo moramo posvetiti merjenju razdalj:
  - med **posameznim učnim primerom** in **gručo**
  - med **dvema gručama**
- kot razdaljo v teh primerih lahko upoštevamo:



- razdaljo med **najbližjima** primeroma (enojna povezanost, angl. *single linkage*)  
$$d(C_1, C_2) = \min_{i,j} \{d_{ij} | i \in C_1, j \in C_2\}$$
- razdaljo med **najbolj oddaljenima** primeroma (popolna povezanost, angl. *complete linkage*)  
$$d(C_1, C_2) = \max_{i,j} \{d_{ij} | i \in C_1, j \in C_2\}$$
- **povprečno razdaljo** med vsemi primeri (povprečna povezanost, angl. *average linkage*)  
$$d(C_1, C_2) = \sum_{i \in C_1, j \in C_2} \frac{d_{ij}}{|C_1||C_2|}$$

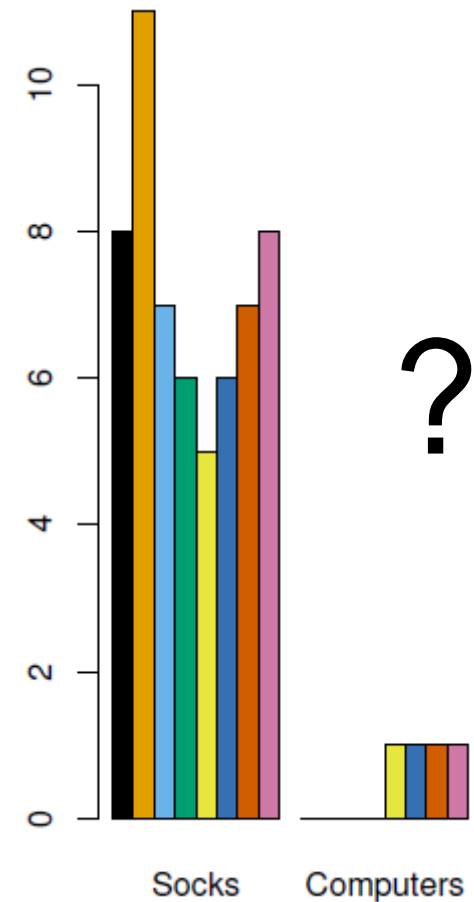
# Primer

- 45 primerov, 2 atributa (oznaka razreda je skrita pred algoritmom za gručenje)
- uporabljena evklidska razdalja in merjenje razdalj s polno povezanostjo (complete linkage)
- dendrogram prikazuje rezanja na različnih višinah



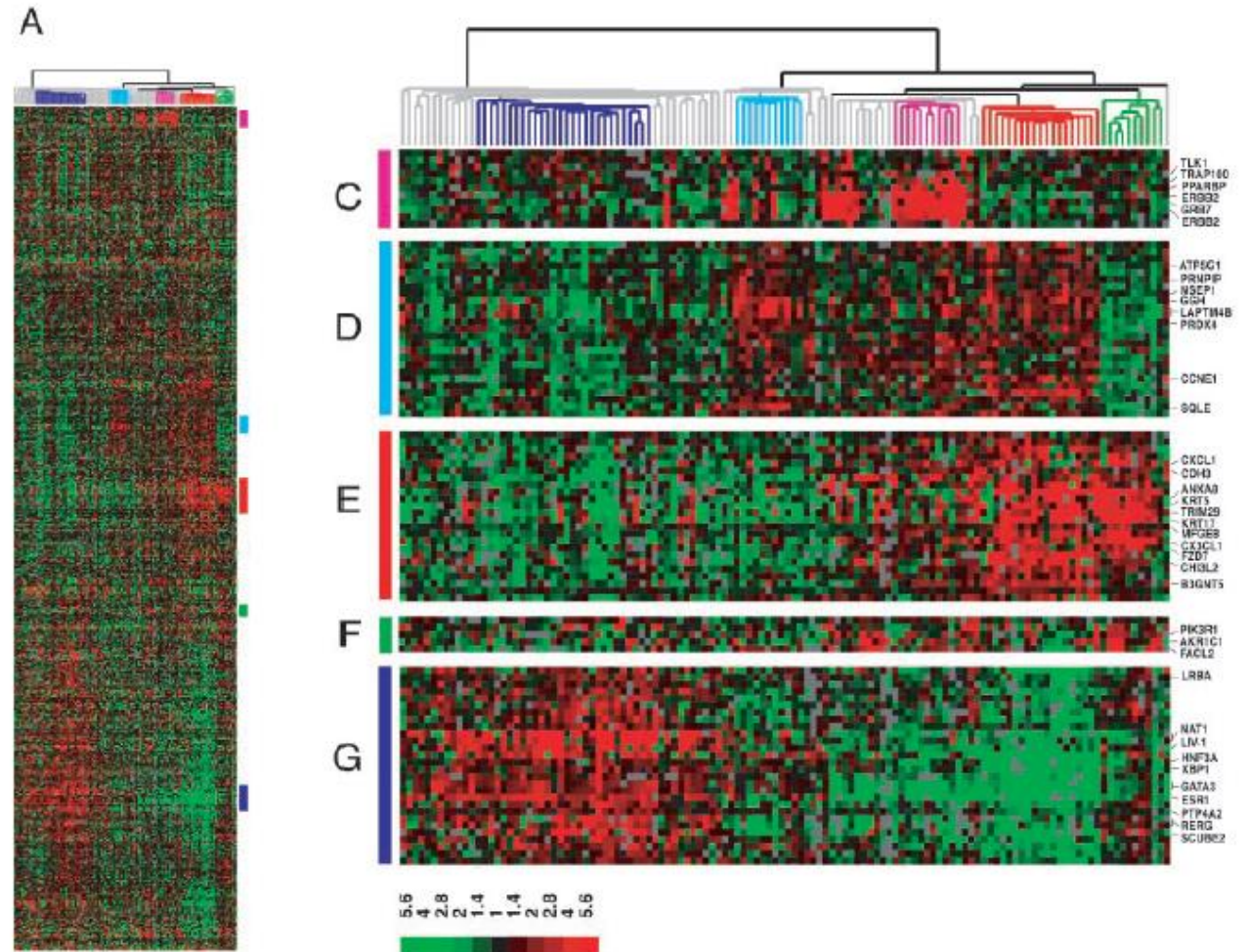
# Opombe

- **normalizacija atributov** (glej desno sliko →)
- **časovna zahtevnost:**
  - združevalni pristop:  $O(n^2 \log n)$ :  
 $n^2$  časa za izračun matrike razdalj,  $\log n$  za urejanje razdalj
  - delilni pristop:  $O(2^n)$ :  
za iskanje optimalne delitve na dve podgruči
- **parametri**
  - katero mero razdalje izbrati?
  - kateri pristop merjenja razdalj med gručkami izbrati?
  - kolikšno naj bo ciljno število gručk?



# Primer uporabe

- analiza različnih vrst tumorja na prsih glede na izraženost genov
  - Sørli, Therese, et al. "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications." Proceedings of the National Academy of Sciences 98.19 (2001): 10869-10874.
- mera razdalje: korelacija
- razdalja med gručkami: povprečna razdalja med primeri
- atributi: izraženost 500 genov
- rezultati:
  - identifikacija sorodnih skupin pacientov
  - identifikacija izraženih genov v skupinah pacientov



# Izpitna naloga

- 2. izpitni rok, 13. 2. 2019 (prilagojena naloga)

## 4. NALOGA (10t):

Podanih je pet točk z vrednostmi atributov X in Y, ki predstavljata koordinati na grafu.

- a) (5t) Izvedi algoritem hierarhičnega razvrščanja naštetih točk in nariši dendrogram. Uporabi Manhattansko razdaljo in pristop popolne povezanosti (angl. complete linkage) merjenja razdalj med gruči.
- b) (3t) Dendrogram iz prejšnje naloge poreži tako, da dobimo dve gruči.

točka	X	Y
A	1	1
B	3	1
C	1	3
D	3	2
E	4	3

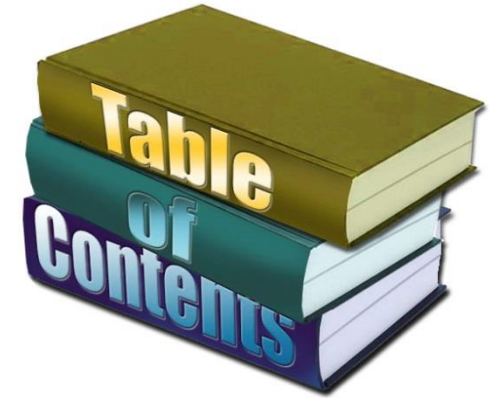
# Pregled

## I. strojno učenje

- uvod v strojno učenje
- učenje odločitvenih dreves
- učenje dreves iz šumnih podatkov (rezanje dreves)
- ocenjevanje učenja
- diskretizacija atributov, obravnava manjkajočih vrednosti
- naivni Bayesov klasifikator
- nomogrami
- k najbližjih sosedov
- lokalna utežena regresija
- regresijska drevesa
- linearni modeli
- hierarhično gručenje
- gručenje z metodo k voditeljev

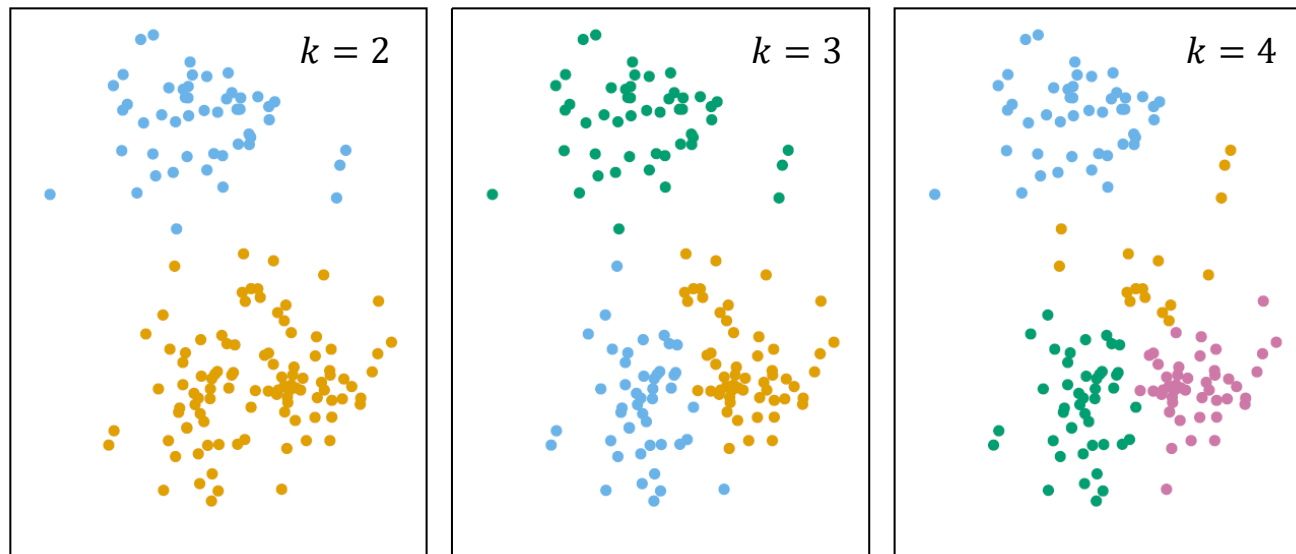
## II. preiskovanje

- definicija in cilji področja
- primeri umetnih in realnih problemov



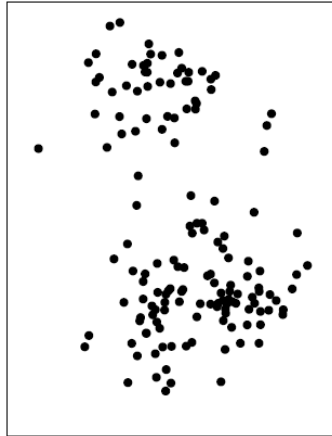
# Metoda voditeljev

- angl. *k-means clustering*
- postopek:
  - izberi **začetno število gruč  $k$**  in **naključno priredi** vsak učni primer eni od gruč
  - ponavljaj do konvergence (dokler se pripadnost gručam spreminja):
    - za vsako izmed  $k$  gruč izračunaj **centroid** (točka, določena s srednjo vrednostjo atributov vseh primerov, ki pripadajo gruči)
    - spremeni pripadnost vsakega primera tisti gruči, katere centroid je najbližji (glede na izbrano mero razdalje)
- primer: različne rešitve za  $k = 2$ ,  $k = 3$  in  $k = 4$

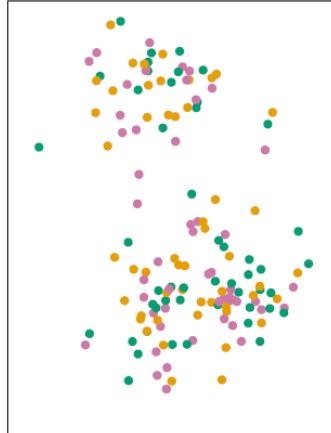


# Primer izvajanja

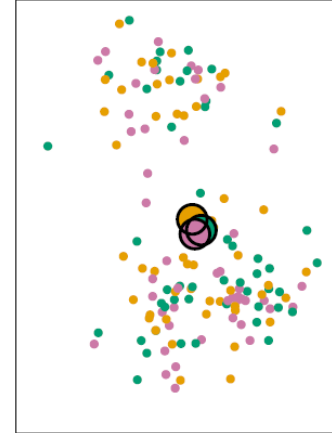
učna množica



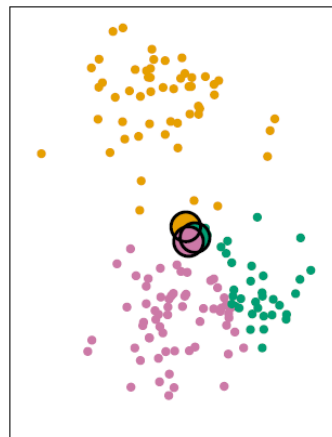
naključna določitev gručam



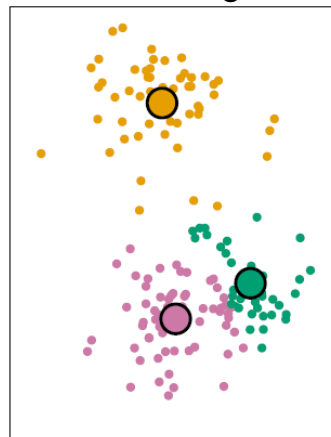
izračun začetnih centroidov gruč



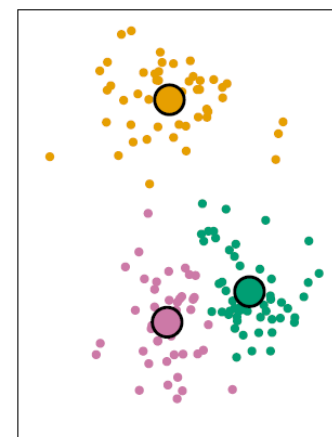
določitev pripadnosti primerov najbližjemu centroidu



ponovni izračun centroidov gruč



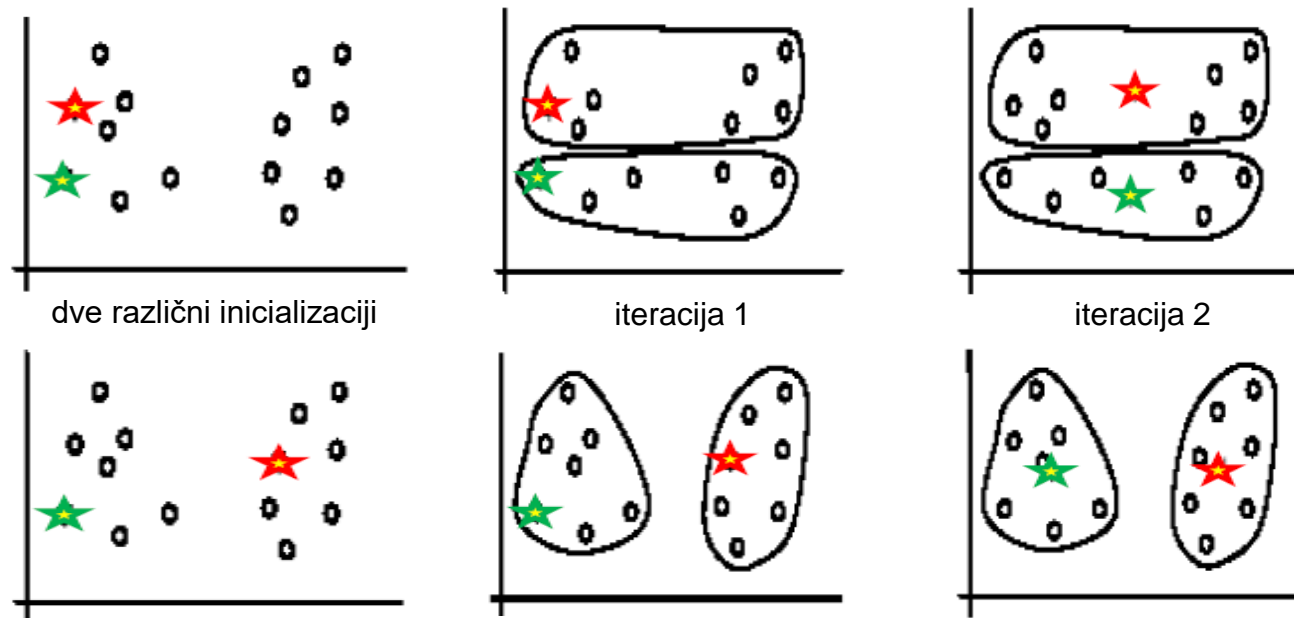
rezultat po 10 iteracijah





# Lastnosti algoritma

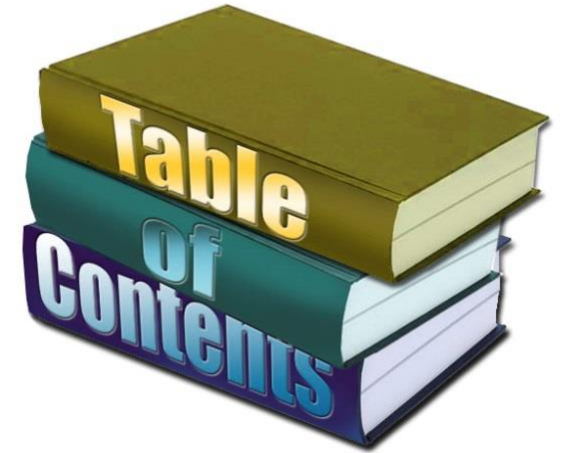
- metoda na vsakem koraku znižuje varianco znotraj gruč (s prerazporejanjem pripadnosti primerov najbližji gruči)
- algoritem ne najde globalnega optimuma (glede na cilj minimizacije variance znotraj gruč), rešitev je odvisna od začetne inicializacije
- izračun centroidov je potrebno prilagoditi, če so atributi diskretni
- algoritem je občutljiv na šum (angl. *outliers*)
- težka objektivna evalvacija, vendar uporabno v praksi



## **II. PREISKOVANJE**

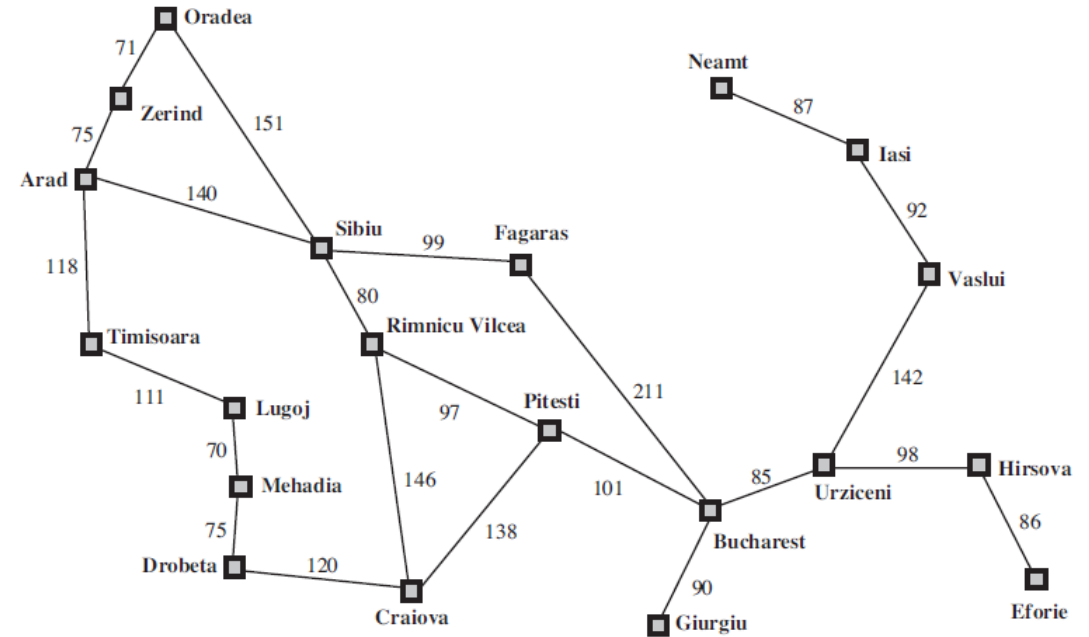
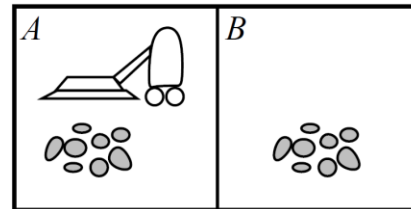
# Pregled

- preiskovanje prostora stanj
  - definicija in cilji področja
  - primeri umetnih in realnih problemov

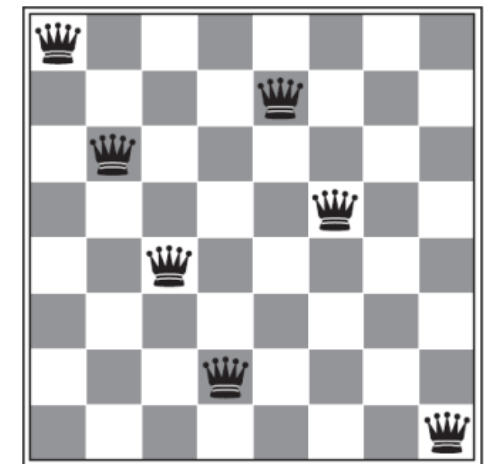
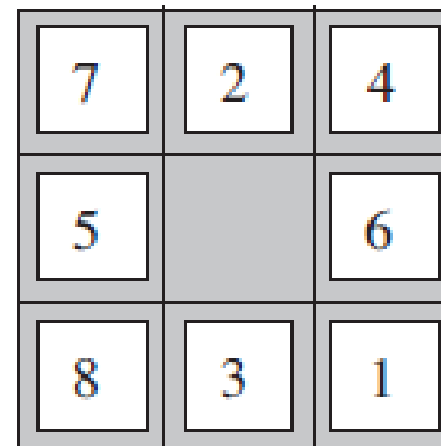


# Primeri problemov

- avtomatski sesalec
- iskanje najkrajše poti
- igra 8 ploščic
- 8 kraljic na šahovnici
- planiranje v svetu kock
- manevriranje robotov
- trgovski potnik



- potrebujemo:
  1. način za abstrakcijo problema
  2. simbolično predstavitev (graf, drevo)
  3. uporabo algoritma na simbolični predstavitvi za iskanje poti do cilja



# Definicija problema

Za formalni opis problema potrebujemo:

- **začetno stanje**
- opis vseh možnih **akcij**, ki so na razpolago v posameznih stanjih
- **prehodno funkcijo**, ki definira naslednika stanja  
$$\text{rezultat}(\text{trenutno\_stanje}, \text{akcija}) = \text{novo\_stanje}$$
- **ciljni predikat**:  
$$\text{cilj}(\text{stanje}) = \{\text{true}, \text{false}\}$$
- **cenilna funkcija**, ki vsaki poti določi ceno (numerično vrednost)  
$$c(\text{stanje}, \text{akcija}, \text{stanje}') \geq 0$$

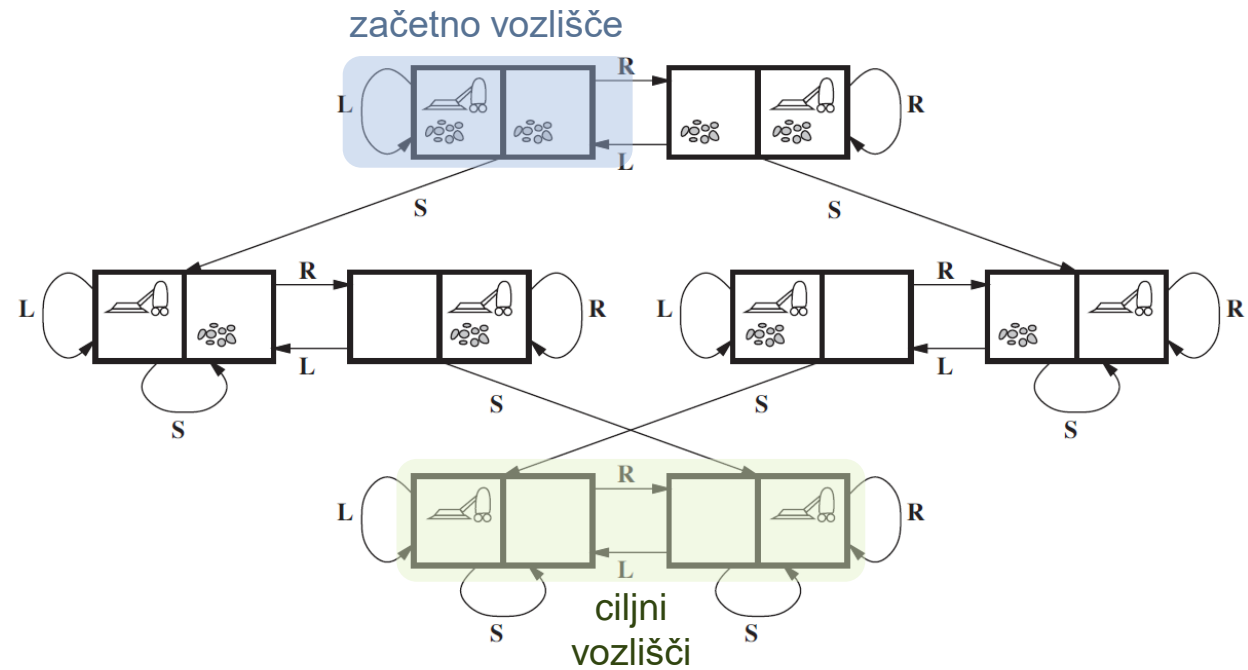


# Definicija problema

- opisani način podajanja problema omogoča, da problem predstavimo z usmerjenim grafom, ki ponazarja **prostor stanj**:
  - **vozlišča**: stanja (problemske situacije)
  - **povezave**: akcije (dovoljene poteze)
  - **pot** v grafu: zaporedje stanj, ki ga povezuje zaporedje akcij
  - eno **začetno** vozlišče
  - eno ali več **ciljnih** vozlišč
- reševanje problema – **iskanje poti v grafu s preiskovanjem**
- **rešitev problema** – zaporedje akcij (pot), ki vodi od začetnega do ciljnega vozlišča
  - optimalna rešitev problema izmed vseh možnih ima najnižjo ceno poti

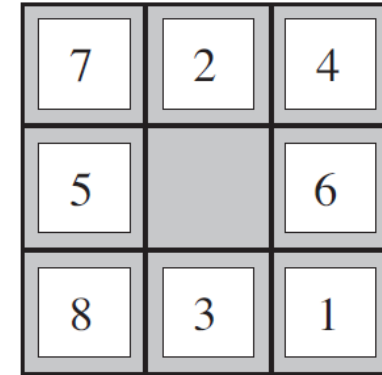
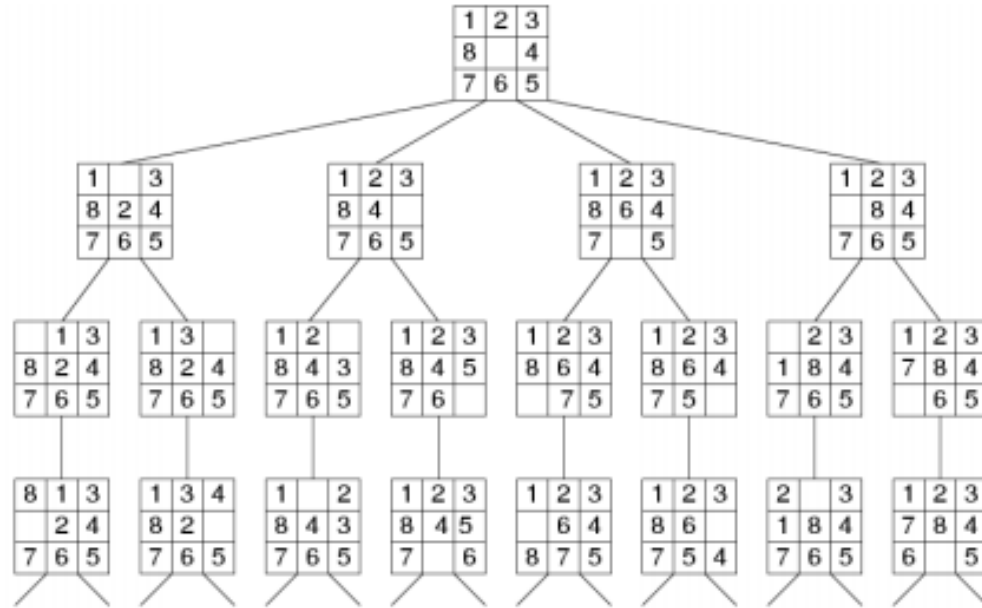


# Primer: sesalec

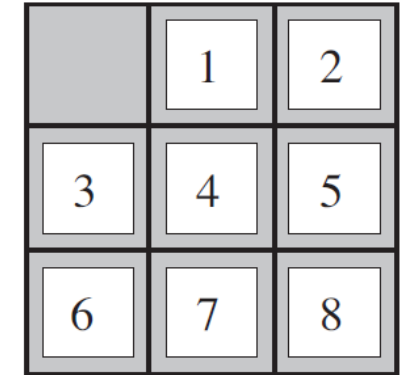


- **stanja** – kombinacija lokacije robota in umazanije v prostorih A in B ( $2 \times 2^2 = 8$  stanj)
- **akcije** – levo, desno, sesaj (in počakaj?)
- **cilj** – oba prostora čista
- **cena akcije** – npr. 1 za vsako potezo

# Primer: igra 8 ploščic



začetno stanje

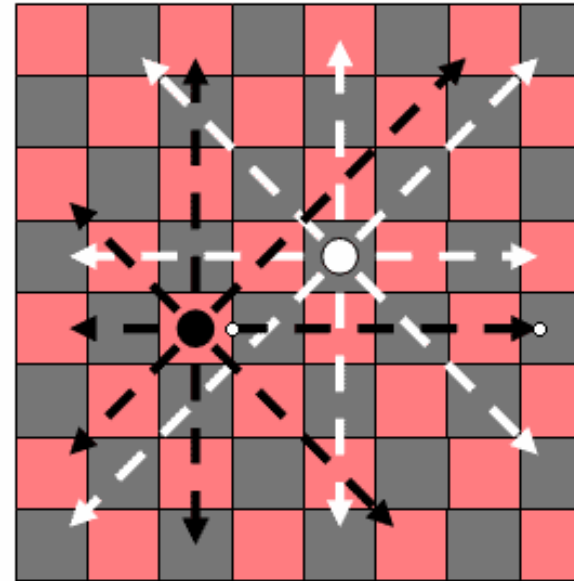
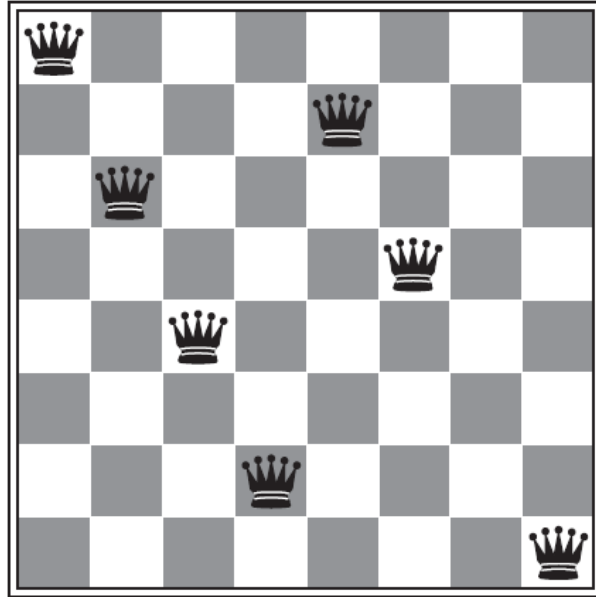


končno stanje

- **stanja** – lokacije ploščic
- **akcije** – premiki "prazne" ploščice levo, desno, gor, dol
- **cilj** – podano ciljno stanje
- **cena akcije** – 1 za vsako potezo
- problem je NP-poln
- igra 3x3 ima  $9!/2 = 181.440$  dosegljivih stanj
- igra 4x4 ima okoli  $1,3 \times 10^{12}$  dosegljivih stanj
- igra 5x5 ima okoli  $10^{25}$  dosegljivih stanj



# Primer: problem 8 kraljic



- **stanja** – postavitve 0-8 kraljic na plošči ( $64 \times 63 \times \dots \times 57 = 1.8 \times 10^{14}$ ) stanj
- **začetno stanje** – prazna igralna plošča
- **akcije** – postavitve nove kraljice na prazno polje
- **cilj** – 8 kraljic na plošči, nobeni dve se ne napadata

bolj optimalna formulacija problema:

- stanja – postavitve 0-8 kraljic od leve proti desni, v vsak stolpec ena (2057 stanj)
- akcije – postavitve nove kraljice na prazno polje v prvi skrajno levi prosti stolpec

# Primer: Knuthov neskončni prostor

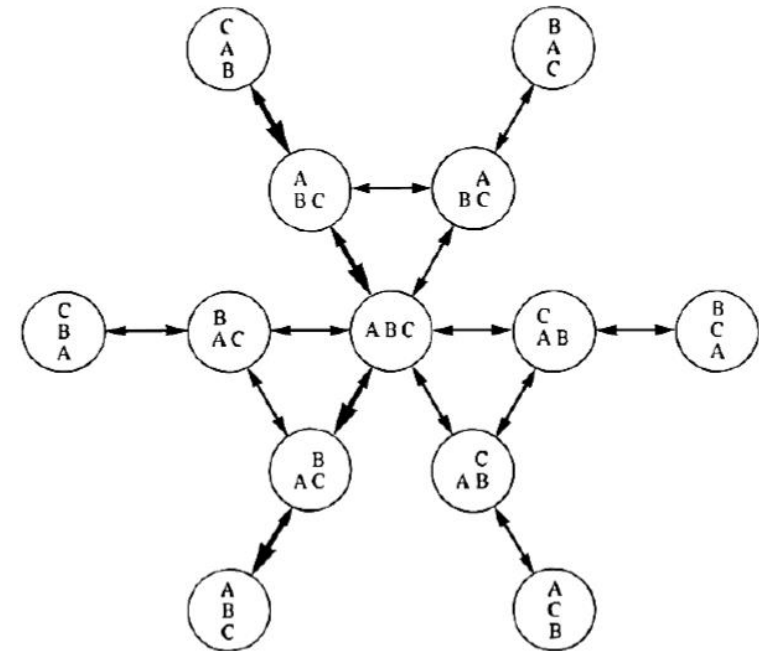
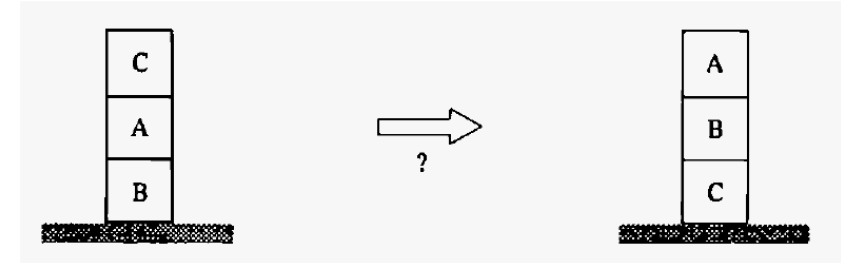
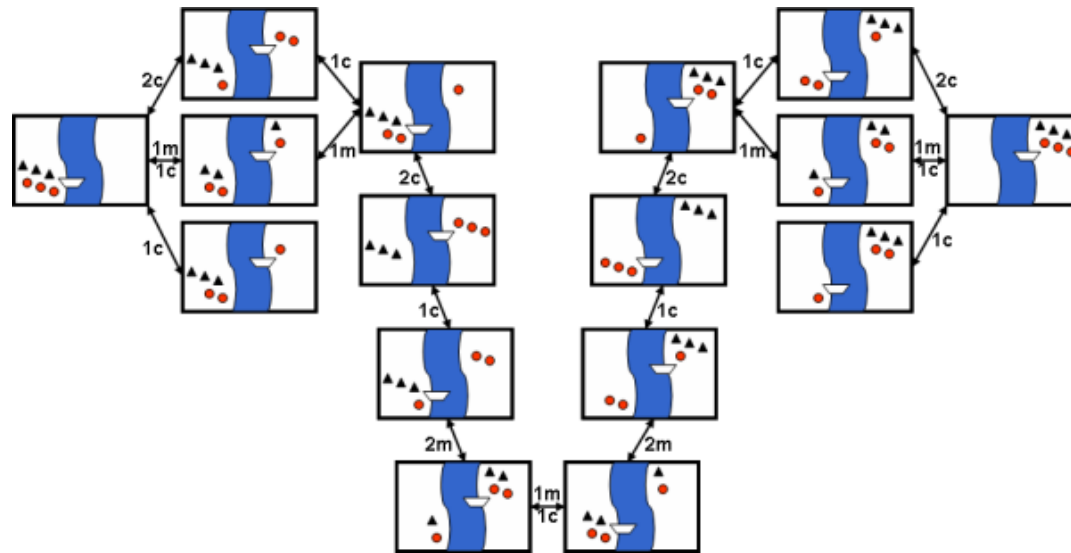
- Knuthova predpostavka: začnši s številom 4 in zaporedjem treh operacij (faktoriela, kvadratni koren, zaokroževanje navzdol), lahko dosežemo poljubno pozitivno celo število. Npr. število 5 lahko zapišemo kot:

$$5 = \left[ \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right]$$

- **začetno stanje** – število 4
- **akcije** – uporaba ene izmed operacij *{faktoriela, kvadratni koren, zaokroževanje navzdol}*
- **stanja** – pozitivna števila
- **ciljno stanje** – želeno pozitivno število

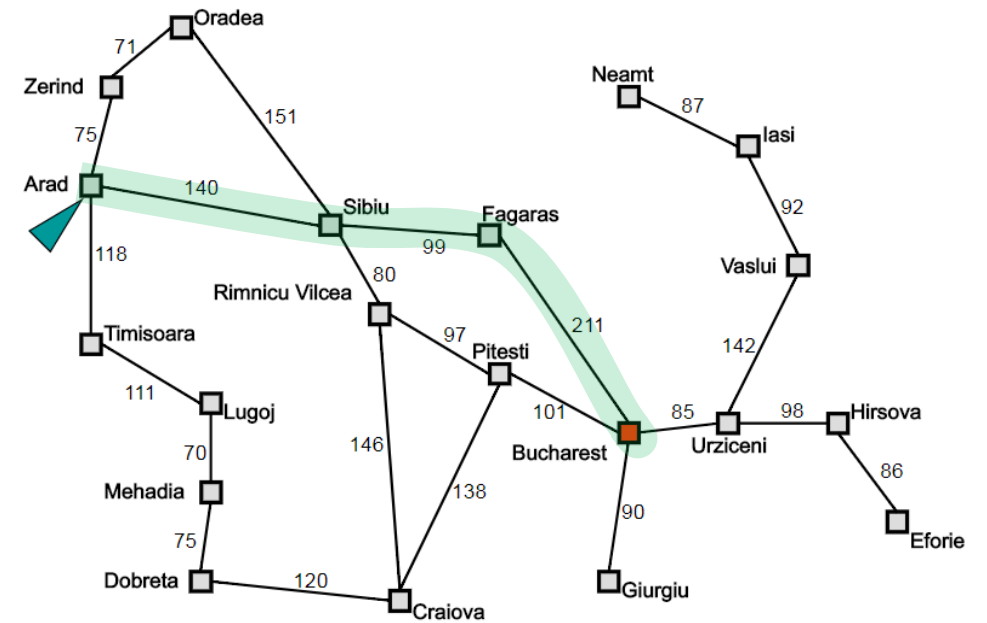
# Primeri drugih umetnih problemov

- misijonarji in ljudožerci
- preurejanje postavitve kock



# Primeri nekaterih realnih problemov

- iskanje najkrajše poti
- primer obhoda ("obišči vsa mesta vsaj enkrat, začeni v mestu X")
- problem potujočega trgovca - TSP ("najdi najkrajšo pot za obhod vseh mest natanko enkrat, začeni v mestu X")
- postavitve komponent na vezju (minimizacija površine, dolžine povezav itd.)
- navigacija robotov v prostoru
- sestavljanje izdelkov na proizvodni liniji



# Preiskovanje

- problem: velika kombinatorična eksplozija možnih stanj
- preiskovalni algoritmi:
  - **neinformirani**: razpolagajo samo z definicijo problema
    - iskanje v širino (*breadth-first search*)
    - iskanje v globino (*depth-first search*)
    - iterativno poglobljanje (*iterative deepening*)
    - cenovno-optimalno iskanje (*uniform-cost search*)
  - **informirani**: razpolagajo tudi z dodatno informacijo (domensko znanje, hevristične ocene), kako bolj učinkovito najti rešitev
    - algoritem A\*
    - algoritem IDA\*
    - prioritetno preiskovanje (*best-first search*)
    - algoritem RBFS (*recursive best-first search*)
    - plezanje na hrib (*hill climbing*)
    - iskanje v snopu (*beam search*)
    - ...



**Preiskovanje prostora stanj,  
neinformirani algoritmi**