

Úvod do umělé inteligence, řešení problémů

Aleš Horák

E-mail: hales@fi.muni.cz
<http://nlp.fi.muni.cz/uui/>

Obsah:

- Co je “umělá inteligence”
- Organizace předmětu PB016
- Řešení problémů

Co je “umělá inteligence”

?

Co je “umělá inteligence”

?

→ `sli.do/uui`

Co je “umělá inteligence”

Communication on AI for Europe, COM 2018/237 final

Artificial intelligence (AI) refers to **systems** that display intelligent behaviour by **analysing** their environment and taking **actions** – with some degree of **autonomy** – to achieve specific **goals**. AI-based systems can be purely **software**-based, acting in the **virtual world** (e.g. voice assistants, image analysis software, search engines, speech and face recognition systems) or AI can be embedded in **hardware** devices (e.g. advanced robots, autonomous cars, drones or Internet of Things applications).

High Level Expert Group, A definition of AI, 2019

Artificial intelligence (AI) systems are **software** (and possibly also **hardware**) **systems** designed by humans that, given a complex **goal**, act in the **physical** or **digital dimension** by perceiving their **environment** through **data** acquisition, interpreting the collected structured or unstructured data, **reasoning** on the knowledge, or processing the **information**, derived from this data and deciding the best **action(s)** to take to achieve the given goal. AI systems can either use symbolic rules or learn a numeric model, and they can also **adapt** their behaviour by analysing how the environment is affected by their previous actions.

Co je “umělá inteligence”

Communication on AI for Europe, COM 2018/237 final

Artificial intelligence (AI) refers to **systems** that display intelligent behaviour by **analysing** their environment and taking **actions** – with some degree of **autonomy** – to achieve specific **goals**. AI-based systems can be purely **software**-based, acting in the **virtual world** (e.g. voice assistants, image analysis software, search engines, speech and face recognition systems) or AI can be embedded in **hardware** devices (e.g. advanced robots, autonomous cars, drones or Internet of Things applications).

High Level Expert Group, A definition of AI, 2019

Artificial intelligence (AI) systems are **software** (and possibly also **hardware**) **systems** designed by humans that, given a complex **goal**, act in the **physical** or **digital dimension** by perceiving their **environment** through **data** acquisition, interpreting the collected structured or unstructured data, **reasoning** on the knowledge, or processing the **information**, derived from this data and deciding the best **action(s)** to take to achieve the given goal. AI systems can either use symbolic rules or learn a numeric model, and they can also **adapt** their behaviour by analysing how the environment is affected by their previous actions.

Co je “umělá inteligence”

- systém, který se chová jako člověk

Co je “umělá inteligence”

- systém, který se chová jako člověk

Turingův test (1950) zahrnuje:

- zpracování přirozeného jazyka (NLP)
- reprezentaci znalostí (KRepresentation)
- vyvozování znalostí (KReasoning)
- strojové učení
- (počítačové vidění)
- (robotiku)

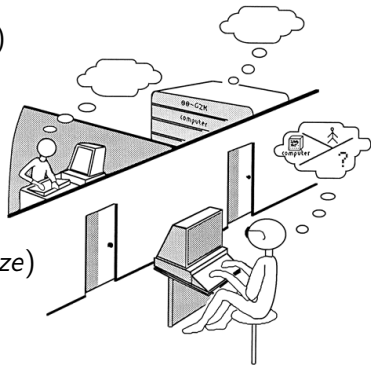
od 1991 – **Loebnerova cena** (*Loebner Prize*)


→ každý rok

\$4.000 za “nejlidštější” program, nabízí


\$100.000 a zlatá medaile za složení celého

Turingova testu (od 2019 bez odměn)



A comic panel showing two men sitting at a table with two mugs of beer. The man on the right is smoking a cigarette. A large white speech bubble originates from the man on the left.


Můj počítač udělal ten Turingův test.

A comic panel showing the same two men at the table. A large white speech bubble originates from the man on the right.

To znamená, že je stejně inteligentní jako já.

A comic panel showing the same two men at the table. A smaller white speech bubble originates from the man on the right.

Nebo ty.

A comic panel showing the same two men at the table. A large white speech bubble originates from the man on the left.

No on to zas tak velkej úspěch není.

- systém, který myslí jako člověk
 - snaha porozumět postupům lidského myšlení – kognitivní (poznávací) věda
 - využívá poznatků neurologie, neurochirurgie, . . .

- systém, který myslí jako člověk

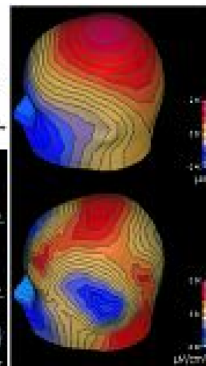
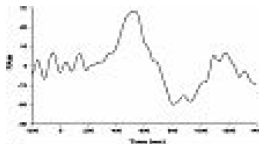
- snaha porozumět postupům lidského myšlení – **kognitivní (poznávací) věda**
- využívá poznatků neurologie, neurochirurgie, ...
např.

Angela Friederici:

*Language Processing in
the Human Brain*

*Max Planck Institute of
Cognitive Neuroscience,
Leipzig*

měření "Event Related Potentials" (ERP) v mozku –
jako potvrzení oddělení
syntaxe a sémantiky při
zpracování věty



- 2013–2023 **Human Brain Project**, Geneva, Švýcarsko

- systém, který myslí rozumně
 - od dob Aristotela (350 př.n.l.)
 - náplň studia **logiky**
 - problém – umět najít řešení teoreticky × prakticky (složitost a vyčíslitelnost)
 - problém – neúplnost a nejistota vstupních dat

- systém, který myslí rozumně
 - od dob Aristotela (350 př.n.l.)
 - náplň studia **logiky**
 - problém – umět najít řešení teoreticky × prakticky (složitost a vyčíslitelnost)
 - problém – neúplnost a nejistota vstupních dat
- systém, který se chová rozumně (inteligentně)
inteligentní **agent** – systém, který
 - jedná za nějakým účelem
 - jedná samostatně
 - jedná na základě vstupů ze svého prostředí
 - pracuje delší dobu
 - adaptuje se na změny

Čím se budeme zabývat?

- základní **struktury** a **algoritmy** běžně používané při technikách **programování pro inteligentní agenty**
- **strategie** řešení, **prohledávání** stavového prostoru, **heuristiky**, ...
- s příklady ve cvičení v jazyce **Python**

Náplň předmětu

- 1 úvod do UI, řešení problémů
- 2 prohledávání stavového prostoru
- 3 dekompozice problému, problémy s omezujícími podmínkami
- 4 hry a základní herní strategie
- 5 logický agent, výroková logika
- 6 pravdivost, dokazatelnost, důkazové metody
průběžná písemka
- 7 logika prvního řádu a intenzionální logika
- 8 rezoluční metoda, úvod do logického programování
- 9 modální logiky, vícehodnotové logiky
- 10 reprezentace a vyvozování znalostí
- 11 učení, rozhodovací stromy, neuronové sítě
- 12 zpracování přirozeného jazyka

Obsah

- 1 Co je “umělá inteligence”
 - Čím se budeme zabývat?
- 2 Organizace předmětu PB016
 - Základní informace
 - Cvičení
- 3 Řešení problémů
 - Problém osmi dam
 - Problém osmi dam I
 - Problém osmi dam II
 - Problém osmi dam III
 - Další příklad – posunovačka
 - Reálné problémy řešitelné prohledáváním

Organizace předmětu PB016

Hodnocení předmětu:

- dotazníky na cvičení ($11 \times 2 = \text{max } 22$ bodů)
nutná podmínka k závěrečné zkoušce ≥ 8 bodů
- průběžná písemka (max 20 bodů)
 - v 1/2 semestru – v týdnu po 6. přednášce, pro všechny jediný termín
- závěrečná písemka (max 58 bodů)
 - dva řádné a jeden opravný termín
- hodnocení – součet bodů (max 100 bodů)
- známka A za ≥ 92 bodů známka E za ≥ 64 bodů
- rozdíly zk, k, z – různé limity
- někteří mohou získat extra body ve cvičení (max 1/cvičení)
 - oprava chyby v materiálech
 - samostatná aktivita nad rámec zadání
 - nadprůměrné elegantní řešení

Základní informace

- web stránka předmětu – <http://nlp.fi.muni.cz/uui/>
- slajdy i cvičení – průběžně doplňovány v [interaktivních osnovách](#)
- kontakt na přednášející – Aleš Horák <hales@fi.muni.cz>, Luboš Popelínský <popel@fi.muni.cz> (**Subject: PB016 ...**)
- literatura:
 - Russell, S. a Norvig, P.: [Artificial Intelligence: A Modern Approach](#), 3rd ed., Prentice Hall, 2010. (prezenčně v knihovně)
 - slajdy a cvičení na webu předmětu

Cvičení

- materiály viz [interaktivní osnova](#) cvičení
- probíhat budou formou online událostí na [univerzitním Google Meet](#)
 - každý obdrží emailem pozvánku k události od příslušného cvičícího
 - jakmile potvrdíte účast, měla by událost být viditelná ve vašem Google kalendáři
 - pro připojení se ke cvičení stačí kliknout na Google Meet odkaz spojený s událostí
- platforma pro řešení úkolů je [Google Colaboratory](#) na sdíleném Google disku
 - Jupyter notebooky (Python + Markdown/LaTeX)
- cvičení jsou [povinná](#) (možné max 3 neomluvené absence)

Obsah

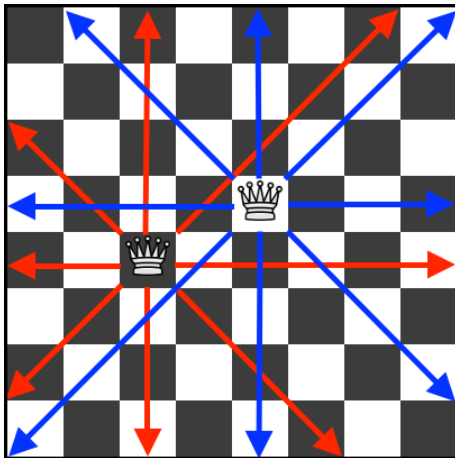
- 1 Co je “umělá inteligence”
 - Čím se budeme zabývat?
- 2 Organizace předmětu PB016
 - Základní informace
 - Cvičení
- 3 Řešení problémů
 - Problém osmi dam
 - Problém osmi dam I
 - Problém osmi dam II
 - Problém osmi dam III
 - Další příklad – posunovačka
 - Reálné problémy řešitelné prohledáváním

Problém osmi dam

úkol: *Rozestavte po šachovnici 8 dam tak, aby se žádné dvě vzájemně neohrožovaly.*

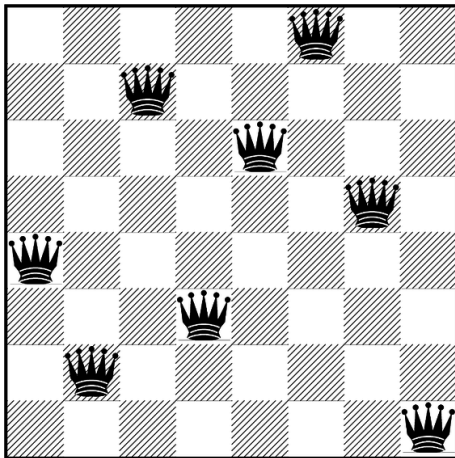
Problém osmi dam

úkol: Rozestavte po šachovnici 8 dam tak, aby se žádné dvě vzájemně neohrožovaly.



Problém osmi dam

úkol: Rozestavte po šachovnici 8 dam tak, aby se žádné dvě vzájemně neohrožovaly.



celkem pro 8 dam existuje 92 různých řešení

Problém osmi dam I

datová struktura – osmiprvková množina

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8)\}$$

Problém osmi dam I

datová struktura – osmiprvková množina

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8)\}$$

`solution = {(1,4), (2,2), (3,7), (4,3), (5,6), (6,8), (7,5), (8,1)}`

Problém osmi dam I

datová struktura – osmiprvková množina

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8)\}$$

`solution = {(1,4), (2,2), (3,7), (4,3), (5,6), (6,8), (7,5), (8,1)}`

```
function N-QUEENS( $n \leftarrow 8$ ,  $queens \leftarrow \{\}$ )  
  if length( $queens$ ) =  $n$  then  
    print  $queens$       # řešení  
  else  
    for  $q_x \leftarrow 1$  to  $n$  do  
      for  $q_y \leftarrow 1$  to  $n$  do  
        if NoAttack( $q_x$ ,  $q_y$ ,  $queens$ ) then  
          N-Queens( $n$ ,  $queens + \{(q_x, q_y)\}$ )
```

Problém osmi dam I

datová struktura – osmiprvková množina

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8)\}$$

`solution = {(1,4), (2,2), (3,7), (4,3), (5,6), (6,8), (7,5), (8,1)}`

```
function N-QUEENS( $n \leftarrow 8$ ,  $queens \leftarrow \{\}$ )  
  if length( $queens$ ) =  $n$  then  
    print  $queens$       # řešení  
  else  
    for  $q_x \leftarrow 1$  to  $n$  do  
      for  $q_y \leftarrow 1$  to  $n$  do  
        if NoAttack( $q_x$ ,  $q_y$ ,  $queens$ ) then  
          N-Queens( $n$ ,  $queens + \{(q_x, q_y)\}$ )  
function NOATTACK( $q_x$ ,  $q_y$ ,  $queens$ )  
  for  $q \in queens$  do  
    if  $q_x = q[1]$  or  $q_y = q[2]$  or  $\text{abs}(q[1]-q_x) = \text{abs}(q[2]-q_y)$  then  
      return False  
  return True
```

Problém osmi dam I

datová struktura – osmiprvková množina

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8)\}$$

`solution = {(1,4), (2,2), (3,7), (4,3), (5,6), (6,8), (7,5), (8,1)}`

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens \leftarrow \{\}$ )
  if length( $queens$ ) =  $n$  then
    print  $queens$       # řešení
  else
    for  $q_x \leftarrow 1$  to  $n$  do
      for  $q_y \leftarrow 1$  to  $n$  do
        if NoAttack( $q_x$ ,  $q_y$ ,  $queens$ ) then
          N-Queens( $n$ ,  $queens + \{(q_x, q_y)\}$ )
function NOATTACK( $q_x$ ,  $q_y$ ,  $queens$ )
  for  $q \in queens$  do
    if  $q_x = q[1]$  or  $q_y = q[2]$  or  $\text{abs}(q[1]-q_x) = \text{abs}(q[2]-q_y)$  then
      return False
  return True
N-Queens(8)
{(8,4), (7,2), (6,7), (5,3), (4,6), (3,8), (2,5), (1,1)}
{(7,2), (8,4), (6,7), (5,3), (4,6), (3,8), (2,5), (1,1)}
...

```

Problém osmi dam II

počet možností u řešení I = $64^8 = 281\,474\,976\,710\,656$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 = 178\,462\,987\,637\,760$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

omezení **stavového prostoru** – každá dáma má svůj sloupec

$\{(1, y_1), (2, y_2), (3, y_3), (4, y_4), (5, y_5), (6, y_6), (7, y_7), (8, y_8)\}$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

omezení **stavového prostoru** – každá dáma má svůj sloupec

$[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

omezení **stavového prostoru** – každá dáma má svůj sloupec

$[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]$

počet možností u řešení II = $8^8 = 16\,777\,216$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

omezení **stavového prostoru** – každá dáma má svůj sloupec

$[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]$

počet možností u řešení II = $8 \cdot 7 \cdot 6 \dots \cdot 1 = 40\,320$

Problém osmi dam II

počet možností u řešení I = $64^8 \approx 2.8 \times 10^{14}$

při neopakování pozic $64 \cdot 63 \cdot 62 \dots \cdot 57 \approx 1.8 \times 10^{14}$

omezení **stavového prostoru** – každá dáma má svůj sloupec

$[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]$

počet možností u řešení II = $8 \cdot 7 \cdot 6 \dots \cdot 1 = 40\,320$

```
function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \leftarrow 1$  to  $n$  do
      if NoAttack( $q_y$ ,  $queens_y$ ) then
        N-Queens( $n$ ,  $queens_y + [q_y]$ )
```

```
function NOATTACK( $q_y$ ,  $queens_y$ )
   $q_x = \text{length}(queens_y) + 1$ 
  for  $i \leftarrow 1$  to length( $queens_y$ ) do
    if  $q_y = queens_y[i]$  or  $\text{abs}(i - q_x) = \text{abs}(queens_y[i] - q_y)$  then
      return False
  return True
```

Problém osmi dam III

k souřadnicím x a y \longrightarrow přidáme i souřadnice diagonály u a v

$$u = x - y$$

$$v = x + y$$

$$D_x = [1..8] \quad \longrightarrow \quad D_u = [-7..7]$$

$$D_y = [1..8] \quad \quad \quad D_v = [2..16]$$

Problém osmi dam III

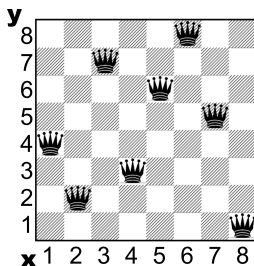
k souřadnicím x a y \longrightarrow přidáme i souřadnice diagonály u a v

$$u = x - y$$

$$v = x + y$$

$$D_x = [1..8] \quad \longrightarrow \quad D_u = [-7..7]$$

$$D_y = [1..8] \quad \longrightarrow \quad D_v = [2..16]$$



Problém osmi dam III

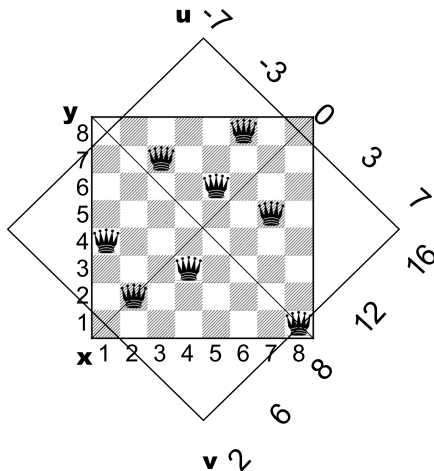
k souřadnicím x a y \longrightarrow přidáme i souřadnice diagonály u a v

$$u = x - y$$

$$v = x + y$$

$$D_x = [1..8] \quad \longrightarrow \quad D_u = [-7..7]$$

$$D_y = [1..8] \quad \quad \quad D_v = [2..16]$$



Problém osmi dam III

po každém umístění dámy aktualizujeme [seznamy volných pozic](#)

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ ,  $d_y \leftarrow []$ ,  $d_u \leftarrow \{\}$ ,  $d_v \leftarrow \{\}$ )
  if length( $queens_y$ ) = 0 and length( $d_y$ ) = 0 then
    return N-Queens( $n$ , [], [1.. $n$ ],  $\{-(n-1)..(n-1)\}$ ,  $\{2..(2*n)\}$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \in d_y$  do
       $q_x \leftarrow$  length( $queens_y$ ) + 1
       $q_u \leftarrow q_x - q_y$ 
       $q_v \leftarrow q_x + q_y$ 
      if  $q_u \in d_u$  and  $q_v \in d_v$  then
        N-Queens( $n$ ,  $queens_y + [q_y]$ ,  $d_y$ .without( $q_y$ ),  $d_u - \{q_u\}$ ,  $d_v - \{q_v\}$ )

```

$d_y = [1..8]$, $d_u = \{-7..7\}$, $d_v = \{2..16\}$

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ ,  $d_y \leftarrow []$ ,  $d_u \leftarrow \{\}$ ,  $d_v \leftarrow \{\}$ )
  if length( $queens_y$ ) = 0 and length( $d_y$ ) = 0 then
    return N-Queens( $n$ , [], [1.. $n$ ],  $\{-(n-1)..(n-1)\}$ ,  $\{2..(2*n)\}$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \in d_y$  do
       $q_x \leftarrow$  length( $queens_y$ ) + 1
       $q_u \leftarrow q_x - q_y$ 
       $q_v \leftarrow q_x + q_y$ 
      if  $q_u \in d_u$  and  $q_v \in d_v$  then
        N-Queens( $n$ ,  $queens_y + [q_y]$ ,  $d_y$ .without( $q_y$ ),  $d_u - \{q_u\}$ ,  $d_v - \{q_v\}$ )

```

$d_y = [1..8]$, $d_u = \{-7..7\}$, $d_v = \{2..16\}$

Problém n dam pro $n = 100$:

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ ,  $d_y \leftarrow []$ ,  $d_u \leftarrow \{\}$ ,  $d_v \leftarrow \{\}$ )
  if length( $queens_y$ ) = 0 and length( $d_y$ ) = 0 then
    return N-Queens( $n$ , [], [1.. $n$ ],  $\{-(n-1) .. (n-1)\}$ ,  $\{2..(2*n)\}$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \in d_y$  do
       $q_x \leftarrow$  length( $queens_y$ ) + 1
       $q_u \leftarrow q_x - q_y$ 
       $q_v \leftarrow q_x + q_y$ 
      if  $q_u \in d_u$  and  $q_v \in d_v$  then
        N-Queens( $n$ ,  $queens_y + [q_y]$ ,  $d_y$ .without( $q_y$ ),  $d_u - \{q_u\}$ ,  $d_v - \{q_v\}$ )

```

$d_y = [1..8], d_u = \{-7..7\}, d_v = \{2..16\}$

Problém n dam pro $n = 100$:
řešení I ... 10^{400}

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ ,  $d_y \leftarrow []$ ,  $d_u \leftarrow \{\}$ ,  $d_v \leftarrow \{\}$ )
  if length( $queens_y$ ) = 0 and length( $d_y$ ) = 0 then
    return N-Queens( $n$ , [], [1.. $n$ ],  $\{-(n-1) .. (n-1)\}$ ,  $\{2..(2*n)\}$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \in d_y$  do
       $q_x \leftarrow$  length( $queens_y$ ) + 1
       $q_u \leftarrow q_x - q_y$ 
       $q_v \leftarrow q_x + q_y$ 
      if  $q_u \in d_u$  and  $q_v \in d_v$  then
        N-Queens( $n$ ,  $queens_y + [q_y]$ ,  $d_y$ .without( $q_y$ ),  $d_u - \{q_u\}$ ,  $d_v - \{q_v\}$ )

```

$d_y = [1..8]$, $d_u = \{-7..7\}$, $d_v = \{2..16\}$

Problém n dam pro $n = 100$:

řešení I ... 10^{400} řešení II ... 10^{158}

Problém osmi dam III

po každém umístění dámy aktualizujeme seznamy volných pozic
počet možností u řešení III = 2057

```

function N-QUEENS( $n \leftarrow 8$ ,  $queens_y \leftarrow []$ ,  $d_y \leftarrow []$ ,  $d_u \leftarrow \{\}$ ,  $d_v \leftarrow \{\}$ )
  if length( $queens_y$ ) = 0 and length( $d_y$ ) = 0 then
    return N-Queens( $n$ , [], [1.. $n$ ],  $\{-(n-1) .. (n-1)\}$ ,  $\{2..(2*n)\}$ )
  if length( $queens_y$ ) =  $n$  then
    print  $queens_y$       # řešení
  else
    for  $q_y \in d_y$  do
       $q_x \leftarrow$  length( $queens_y$ ) + 1
       $q_u \leftarrow q_x - q_y$ 
       $q_v \leftarrow q_x + q_y$ 
      if  $q_u \in d_u$  and  $q_v \in d_v$  then
        N-Queens( $n$ ,  $queens_y + [q_y]$ ,  $d_y$ .without( $q_y$ ),  $d_u - \{q_u\}$ ,  $d_v - \{q_v\}$ )

```

$d_y = [1..8], d_u = \{-7..7\}, d_v = \{2..16\}$

Problém n dam pro $n = 100$:

řešení I ... 10^{400}

řešení II ... 10^{158}

řešení III ... 10^{52}

Další příklad – posunovačka

počáteční stav (např.)

7	2	4
5		6
8	3	1

→ ... →

cílový stav

	1	2
3	4	5
6	7	8

- hra na čtvercové šachovnici $m \times m$ s $n = m^2 - 1$ očíslovanými kameny
- příklad pro šachovnici 3×3 , posunování osmi kamenů (8-posunovačka)
- **stavy** – pozice všech kamenů
- **akce** – “pohyb” prázdného místa

Další příklad – posunovačka

počáteční stav (např.)

7	2	4
5		6
8	3	1

→ ... →

cílový stav

	1	2
3	4	5
6	7	8

- hra na čtvercové šachovnici $m \times m$ s $n = m^2 - 1$ očíslovanými kameny
- příklad pro šachovnici 3×3 , posunování osmi kamenů (8-posunovačka)
- **stavy** – pozice všech kamenů
- **akce** – “pohyb” prázdného místa

☞ **Optimální řešení** obecné n -posunovačky je **NP-úplné**

Počet stavů	u 8-posunovačky	...	$9!/2 = 181\,440$
	u 15-posunovačky	...	10^{13}
	u 24-posunovačky	...	10^{25}

Reálné problémy řešitelné prohledáváním

- hledání cesty z města A do města B
- hledání itineráře, problém obchodního cestujícího
- návrh VLSI čipu
- navigace auta, robota, . . .
- postup práce automatické výrobní linky
- návrh proteinů – 3D-sekvence aminokyselin
- Internetové vyhledávání informací