

**Detekce trojic**

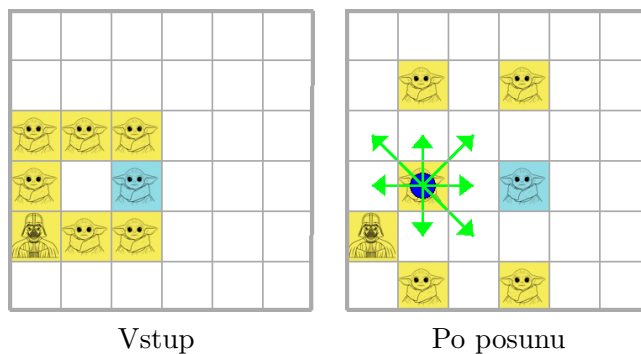
Tento dokument obsahuje zadání lehké i těžké varianty domácího úkolu.

## 1 Zadání lehké varianty

Na šachovnici o rozměrech  $6 \times 6$  jsou figurky Darth Vader a Yoda a každá z nich může mít dvě barvy. Mezi již položené figurky umístíme novou, která způsobí posun okolních figurek. Úkolem programu je v nově vzniklém rozložení figurek detekovat trojice figurek stejné barvy, které leží vedle sebe.

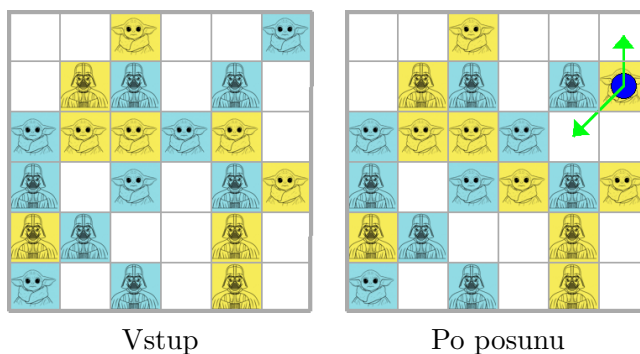
- **Vstup:** parametry na příkazové řádce s významem: `row col figure filename`
  - `row` je číslo řádku pro novou figurku
  - `col` je číslo sloupce pro novou figurku
  - `figure` je označení nově umístěné figurky: -1 (žlutý Yoda), 1 (modrý Yoda), -2 (žlutý Darth Vader) nebo 2 (modrý Darth Vader)
  - `filename` je jméno textového souboru, který obsahuje rozmístění figurek: 6 řádků a na každém řádku je 6 celých čísel oddělených mezerou, kde 0 znamená neobsazenou pozici a čísla (-1,1,-2,2) označují figurky.
- **Výstup:** na standardní výstup vypište souřadnice buněk, kde se vyskytují trojice figurek stejné barvy.
  - Trojice znamená figurky umístěné vedle sebe horizontálně, nad sebou vertikálně, nebo tři v řadě diagonálně. Pokud je trojic více, vypište libovolnou. Pokud žádné trojice nejsou, je výstup **None**.
- Program odevzdejte do Brute pod jménem `detect3.py`
  - Trojice se vypisují ve formátu: `row1 col1 row2 col2 row3 col3` (tři po sobě jdoucí dvojice souřadnic — nejprve řádek a poté sloupec). Dvojice jsou seřazeny vzestupně nejprve podle řádku a poté podle sloupce. Například pokud detekujete trojici na souřadnicích (2,0), (1,1) a (0,2), je třeba tyto dvojice seřadit takto: (0,2) (1,1) (2,0) a vypsat ve formátu: `0 2 1 1 2 0`.
- Velikost šachovnice je vždy  $6 \times 6$ , řádky jsou shora dolů, sloupce zleva doprava, číslujeme od 0.
- Pozice (`row,col`), na kterou se má umístit figurka, je vždy volná. Vstupní soubor vždy existuje a splňuje výše popsaný formát.
- Detekce trojic se provádí až poté, co dojde k posunu figurek dle následujících pravidel. Nově umístěná figurka (umístěná na pozici (`row, col`)) odpuzuje figurky ve svém 8-mi okolí — tj. způsobí jejich posun v jednom ze směrů v 8-mi okolí takto:
  - Yoda odpuzuje jiné figurky Yoda ve svém okolí (bez ohledu na barvu), ale neodpuzuje figurku Darth Vader.
  - Darth Vader odpuzuje figurku Darth Vader i figurku Yoda ve svém okolí (bez ohledu na barvu).
  - K posunu figurky dojde pouze pokud je ve směru posunu volné místo.
  - Pokud je ve směru posunu prostor mimo šachovnici, figurka vypadne.

### 1.1 Příklady posunu figurek



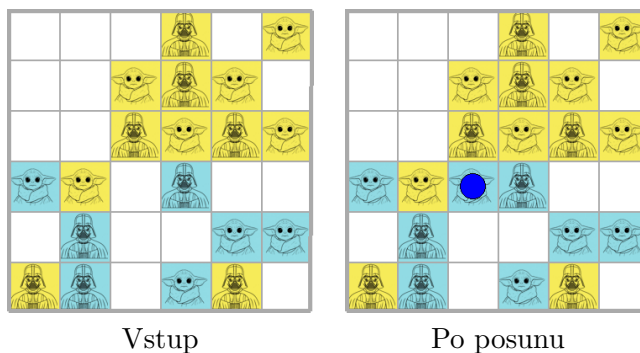
Yoda umístěn na pozici (3,1), způsobí posun všech figurek Yoda ve svém okolí, přičemž figurky na kraji hrací desky (ve sloupci 0) vypadnou. Figurka Darth Vader na pozici (4,0) zůstane beze změny, neboť nereaguje na přítomnost Yoda.

### 1.2 Příklady posunu figurek



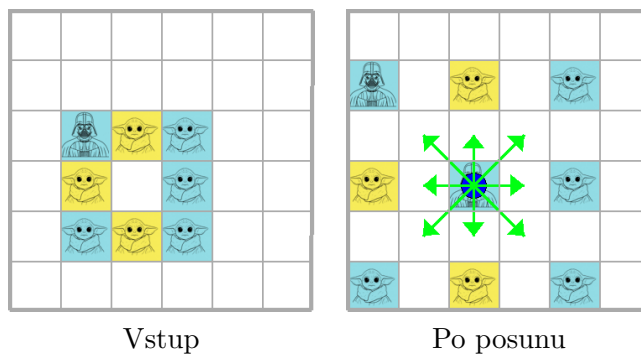
Yoda umístěn na pozici (1,5) a odsune dvě figurky Yoda (naznačeno šipkami). Figurka Darth Vader na pozici (1,4) se nepohne (Yoda nepohne figurkou Darth Vader).

### 1.3 Příklady posunu figurek



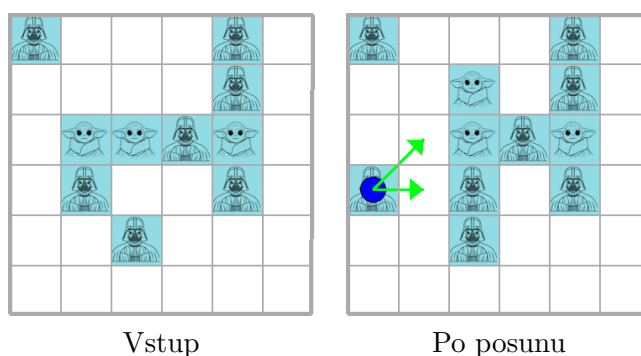
Yoda umístěn na pozici (3,2), ale neodsune žádnou figurku ve svém okolí, protože jejich posun je blokován.

## 1.4 Příklady posunu figurek



Darth Vader umístěn na pozici (3,2), dojde k posunu všech figurek.

## 1.5 Příklady posunu figurek



Darth Vader umístěn na pozici (3,0), dojde k posunu dvou figurek.

## 2 Příklad

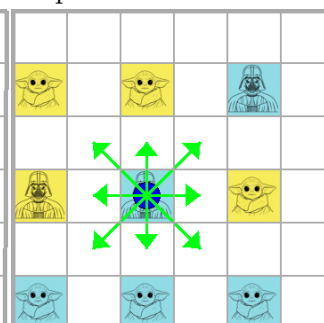
```
python3 detect3.py 3 2 2 situace.txt  
situace.txt:
```

```
0 0 0 0 0 0  
0 0 0 0 0 0  
0 -1 -1 2 0 0  
0 -2 0 -1 0 0  
0 1 1 1 0 0  
0 0 0 0 0 0
```

Figurky v souboru



Po posunutí



Figurky po posunutí (modré pole na pozici 3,2 označuje figurku, která byla nově vložena). Šipky ukazují, které figurky byly posunuty.

Počet trojic: 0

Příklady řešení:

Řešení je: None

### 3 Příklad

```
python3 detect3.py 4 5 -1 situace.txt
situace.txt:
```

```
0 2 2 2 0 0
2 1 1 0 0 2
2 0 1 2 0 0
-2 0 0 0 0 -2
2 0 -1 -1 0 0
1 -1 1 -2 0 0
```

Figurky v souboru



Po posunutí



Figurky po posunutí (modré pole na pozici 4,5 označuje figurku, která byla nově vložena). Šipky ukazují, které figurky byly posunuty.

Počet trojic: 5

Příklady řešení:

Řešení 0/5

```
0 1 0 2 0 3
```



Řešení 1/5

```
0 1 1 2 2 3
```



Řešení 2/5

```
0 2 1 2 2 2
```



Řešení 3/5

0 2 1 1 2 0

Řešení 4/5

1 0 1 1 1 2



## 4 Příklad

python3 detect3.py 5 1 2 situace.txt  
situace.txt:

```
0 0 -1 -1 0 -2
-2 -2 2 2 1 2
0 2 2 0 0 0
0 0 2 2 -2 0
0 1 1 0 -1 0
0 0 2 1 0 2
```

Figurky v souboru

Po posunutí



Figurky po posunutí (modré pole na pozici 5,1 označuje figurku, která byla nově vložena). Šipky ukazují, které figurky byly posunuty.

Počet trojic: 10

Příklady řešení:

Řešení 0/10

1 2 1 3 1 4



Řešení 1/10

1 2 2 2 3 2



Řešení 2/10

1 3 1 4 1 5



Řešení 3/10

1 3 2 2 3 1



Řešení 4/10

2 2 3 2 4 2



Řešení 5/10

3 1 3 2 3 3



Řešení 6/10

3 1 4 2 5 3



Řešení 7/10

3 2 4 2 5 2



Řešení 8/10

3 3 4 2 5 1



Řešení 9/10

5 1 5 2 5 3





## Nalezení nejlepšího tahu

### 5 Zadání těžké varianty úkolu

Uvažujme stejné figurky a pravidla jejich posunu jako v zadání lehké varianty. Na vstupu je počáteční rozmístění figurek, a souřadnice, kam se má vložit nová figurka. Program nejprve určí, jak se změní rozložení figurek (podle pravidel pohybu) a následně hledá nejlepší umístění další figurky tak, aby, po dalším aplikování pravidel pohybu, došlo k vytvoření co největšího počtu trojic stejné barvy, jako má druhá umístěná figurka.

Schematicky program vykoná tuto činnost:

$$s_0 + (r_0, c_0, f_0) \rightarrow \text{pohyb} \rightarrow s_1$$

$$s_1 + (r_1, c_1, f_1) \rightarrow \text{pohyb} \rightarrow s_2,$$

kde  $s_0$  je počáteční stav figurek (ze vstupního souboru),  $s_1$  je stav figurek po vložení figurky  $f_0$  na pozici  $r_0, c_0$  a aplikování pravidel pohybu. V rozložení figurek  $s_1$  program hledá souřadnice  $r_1, c_1$  pro figurku  $f_1$  tak, aby po aplikování pravidel pohybu vzniklo co nejvíc trojic stejné barvy, jako má figurka  $f_1$ . Program tedy hledá trojice v situaci  $s_2$ .

- **Vstup:** parametry na příkazové řádce s významem: `r0 c0 f0 f1 filename`
  - `r0, c0` je souřadnice buňky v počátečním rozmístění figurek  $s_0$  na kterou se vloží figurka  $f_0$ . Poté program aplikuje pohybová pravidla a určí nové rozložení figurek  $s_1$ .
  - `f1` je figurka, pro kterou hledáme nejlepší umístění v novém rozložení figurek  $s_1$ .
  - `filename` je jméno textového souboru, který obsahuje počáteční rozmístění figurek  $s_0$ . Formát souboru je stejný jako v předchozím zadání.
- Figurky, tj. vstupní argumenty `f0` a `f1` nabývají hodnot -2,-1,1,2 (jejich význam je vysvětlen u lehké varianty úkolu).
- **Výstup:** na standardní výstup vypíše dvě celá čísla oddělená mezerou: `r c`
  - Tato souřadnice označuje buňku (řádek, sloupec) v novém rozložení figurek  $s_1$ , na kterou se má položit `f1` aby vzniklo co nejvíc trojic figurek stejné barvy jako má figurka `f1`. Pokud existuje více řešení, vypíše kterékoliv z nich.
- Program odevzdejte do Brute pod jménem `bestmove.py`

#### 5.1 Příklad

```
python3 bestmove.py 0 3 2 2 situace.txt
situace.txt:
```

```
0 0 0 0 0 0
0 0 2 0 0 1
0 2 0 1 0 0
0 0 0 0 0 1
2 0 0 1 1 0
2 0 0 0 0 2
```

Celkový počet řešení: 3.

Figurky v souboru  
stav  $s_0$



Po vložení 2 na pozici 0,3  
stav  $s_1$



Řešení 1/3. Výstup programu

3 0

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 3, 0 je: 3.

stav  $s_2$



Řešení 2/3. Výstup programu

3 3

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 3, 3 je: 3.

stav  $s_2$



Řešení 3/3. Výstup programu

5 3

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 5, 3 je: 3.

stav  $s_2$



## 5.2 Příklad

python3 bestmove.py 0 0 1 2 situace.txt  
situace.txt:

```
0 2 0 0 0 0
1 0 1 0 0 0
2 0 0 0 0 0
2 0 0 0 0 0
0 0 0 0 1 0
0 0 0 0 0 0
```

Celkový počet řešení: 4.

Řešení 1/4. Výstup programu

0 2

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 0, 2 je: 3.

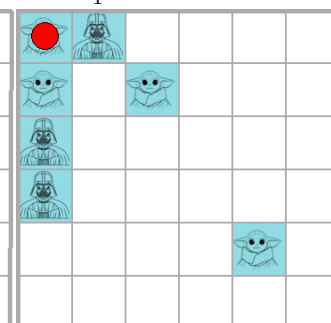
Figurky v souboru

stav  $s_0$

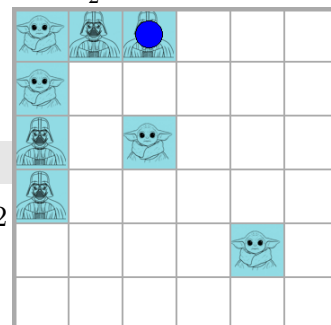


Po vložení 1 na pozici 0,0

stav  $s_1$



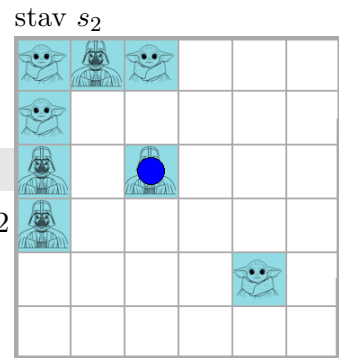
stav  $s_2$



Řešení 2/4. Výstup programu

2 2

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 2, 2 je: 3.



Řešení 3/4. Výstup programu

2 3

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 2, 3 je: 3.



Řešení 4/4. Výstup programu

4 0

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 4, 0 je: 3.



### 5.3 Příklad

python3 bestmove.py 5 2 2 2 situace.txt  
situace.txt:

```
0 0 1 0 0 0
0 0 1 0 0 0
0 0 2 0 2 0
1 0 2 0 0 0
0 0 0 2 0 0
2 0 0 0 0 0
```

Celkový počet řešení: 6.

Řešení 1/6. Výstup programu

1 4

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 1, 4 je: 3.

Figurky v souboru

stav  $s_0$



Po vložení 2 na pozici 5,2

stav  $s_1$



stav  $s_2$



Řešení 2/6. Výstup programu

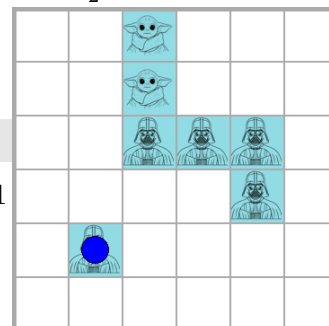
1 5

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 1, 5 je: 3.

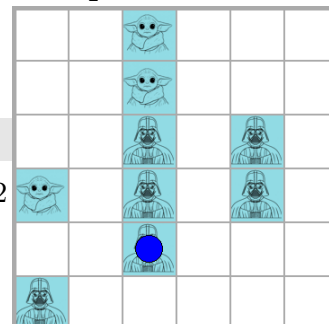
stav  $s_2$



stav  $s_2$



stav  $s_2$



stav  $s_2$



stav  $s_2$



Řešení 3/6. Výstup programu

4 1

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 4, 1 je: 3.

Řešení 4/6. Výstup programu

4 2

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 4, 2 je: 3.

Řešení 5/6. Výstup programu

4 4

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 4, 4 je: 3.

Řešení 6/6. Výstup programu

4 5

Počet trojic, které vzniknou umístěním figurky 2 ve stavu  $s_1$  na pozici 4, 5 je: 3.

## 5.4 Příklad

python3 bestmove.py 2 5 1 1 situace.txt  
situace.txt:

```
0 0 1 0 2 0
0 0 0 0 1 0
0 0 0 0 0 0
0 2 1 0 0 0
2 0 0 0 0 0
0 1 1 0 2 0
```

Celkový počet řešení: 1.

Řešení 1/1. Výstup programu

```
5 3
```

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 5, 3 je: 3.

Figurky v souboru  
stav  $s_0$



Po vložení 1 na pozici 2,5  
stav  $s_1$



stav  $s_2$



## 5.5 Příklad

python3 bestmove.py 0 2 2 1 situace.txt  
situace.txt:

```
0 1 0 -1 0 0
1 -1 -1 0 1 0
1 0 1 0 0 0
1 -1 -1 0 -1 0
-1 1 0 0 0 0
0 1 -1 0 1 -1
```

Celkový počet řešení: 15.

Řešení 1/15. Výstup programu

```
0 3
```

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 0, 3 je: 2.

Figurky v souboru  
stav  $s_0$



Po vložení 2 na pozici 0,2  
stav  $s_1$



stav  $s_2$



Řešení 2/15. Výstup programu

0 5

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 0, 5 je: 2.



Řešení 3/15. Výstup programu

1 3

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 1, 3 je: 2.



Řešení 4/15. Výstup programu

1 5

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 1, 5 je: 2.



Řešení 5/15. Výstup programu

2 3

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 2, 3 je: 2.



Řešení 6/15. Výstup programu

2 4

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 2, 4 je: 2.



Řešení 7/15. Výstup programu

2 5

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 2, 5 je: 2.



Řešení 8/15. Výstup programu

3 3

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 3, 3 je: 2.



Řešení 9/15. Výstup programu

3 5

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 3, 5 je: 2.



Řešení 10/15. Výstup programu

4 2

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 4, 2 je: 2.



Řešení 11/15. Výstup programu

4 3

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 4, 3 je: 2.



Řešení 12/15. Výstup programu

4 4

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 4, 4 je: 2.

stav  $s_2$



stav  $s_2$



stav  $s_2$



stav  $s_2$



Řešení 13/15. Výstup programu

4 5

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 4, 5 je: 2.

Řešení 14/15. Výstup programu

5 0

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 5, 0 je: 2.

Řešení 15/15. Výstup programu

5 3

Počet trojic, které vzniknou umístěním figurky 1 ve stavu  $s_1$  na pozici 5, 3 je: 2.