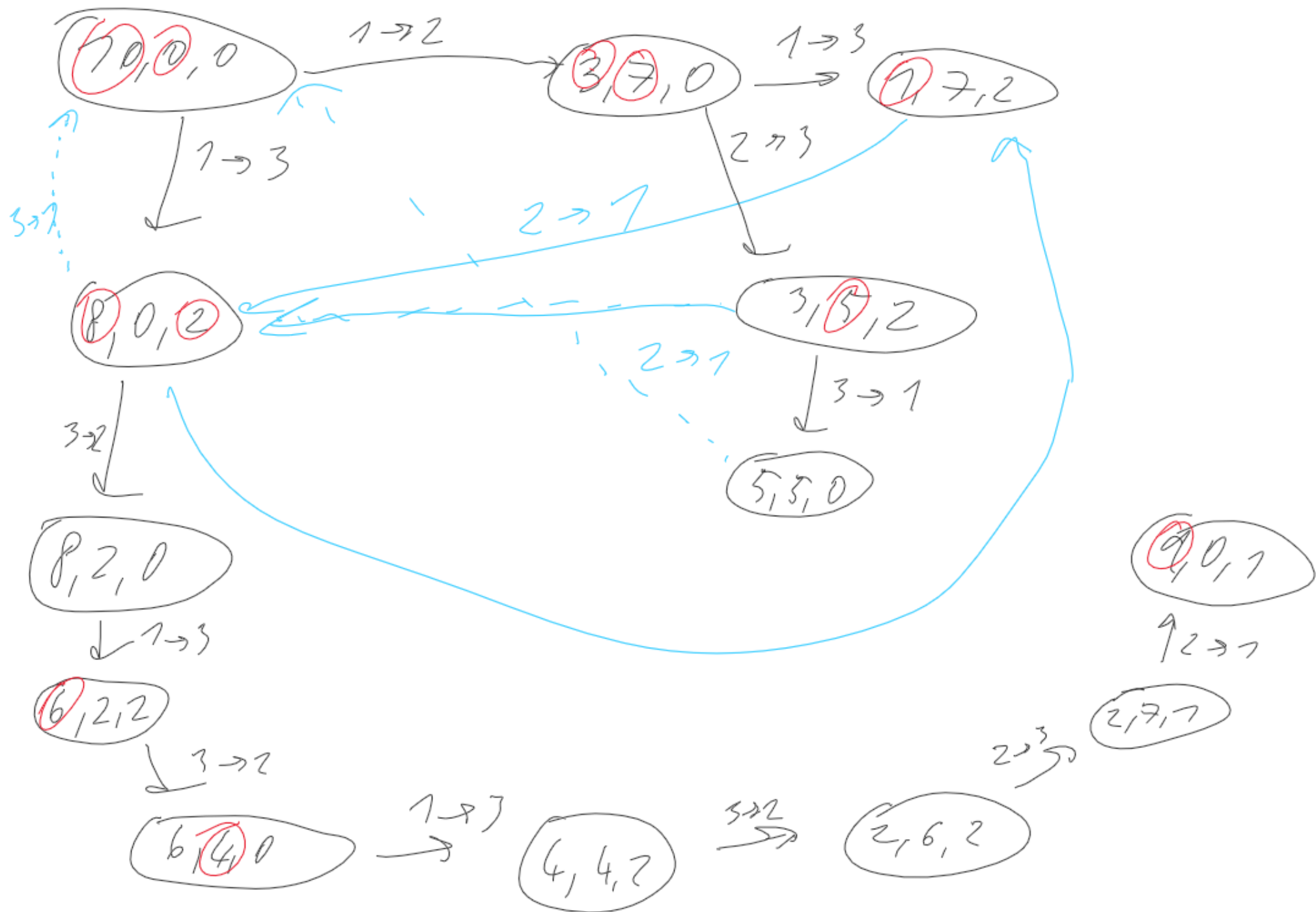


# Algoritmizace a Programování 1

10. cvičení

# Prohledávání stavového prostoru



# Algoritmy

```
prozkoumane = {}
k_prozkoumani = (pocatecni_stav) # TODO jaká datová struktura?

while ...: # TODO kdy skončit?
    stav = k_prozkoumani.get() # konkrétní metoda v závislosti na zvolené DS
    for naslednik in naslednici(stav):
        k_prozkoumani.add(naslednik) # konkrétní metoda v závislosti na zvolené DS
    prozkoumane.add(stav)
```

```
uz_videne = {}

def prozkoumej_stav(stav):
    for naslednik in naslednici(stav):
        if naslednik not in uz_videne:
            prozkoumej_stav(naslednik)
```

Programování – co se nestihlo

# Slovník, množina a vlastní datové typy

- Vytvořte třídu Book s atributy jméno a autor
- Vytvořte 2 instance, které ale reprezentují stejnou knihu
- Použijte je ve slovníku nebo množině, například:

```
already_read = set()
already_read.add(Book("1984", "George Orwell"))
book = Book("1984", "George Orwell")
print(book in already_read)
```

- Chová se to tak, jak chceme?
- Chová se to stejně, jako když použijeme pouze název knihy (string) místo naší třídy?

# Slovník – `dict`, `{}`

- Dvojice klíč-hodnota
- Rychlé operace (průměrně  $O(1)$ )
  - Vložení `slovník[klíč] = hodnota`
  - Získání hodnoty `ulozena_hodnota = slovník[klíč]`
  - Je klíč ve slovníku? `klíč in slovník`
- Klíč musí být hashovatelný
  - ANO: číslo, string, tuple
  - NE: list, dict, set
  - vlastní třída: každá instance je unikátní
    - Lze změnit (metody `__hash__` a `__eq__`)
- Položky nejsou uspořádané!

# Čtení souborů

- Z webu cvičení si stáhněte soubory.zip obsahující dva textové soubory
- Napište program, který je přečte a vypíše s čísly řádek:

```
1: Příliš  
2: žloutoučký kůň
```



Minimax, alfa-beta

```
def alfabeta(vrchol, hloubka,  $\alpha$ ,  $\beta$ )
    if hloubka == 0 or vrchol je koncový:
        return ohodnocení vrcholu
    if hrajeMax(vrchol):
        nej =  $-\infty$ 
        for každé dítě vrcholu:
            nej = max(nej, alfabeta(dítě, hloubka-1,  $\alpha$ ,  $\beta$ ))
             $\alpha$  = max( $\alpha$ , nej)
            if  $\beta \leq \alpha$ :
                break #  $\beta$ -řez
        return nej
    else: # hrajeMin(vrchol)
        nej =  $\infty$ 
        for každé dítě vrcholu:
            nej = min(nej, alfabeta(dítě, hloubka-1,  $\alpha$ ,  $\beta$ ))
             $\beta$  = min( $\beta$ , nej)
            if  $\beta \leq \alpha$ :
                break #  $\alpha$ -řez
        return nej
```

**MAX**

**MIN**

**MAX**

