



# „Algoritmus prohledávání s návratem“ (backtracking)

## Prohledávání do hloubky

Na každém rozcestí, na které přijdeš, zjisti všechny možné cesty a jednu z nich si vyber.

Takto postupuj dokud to jde nebo dokud se nedostaneš do situace, ve které jsi už byl.

Nemůžeš-li dál, vrať se na poslední rozcestí, kde ses rozhodoval – a rozhodni se jinak.

## Příklad: (Theseus, Ariadna a Minotaurus)

### Úloha:

- určitě najít Minotaura (je-li tam)
- po (ne)nalezení se bez bloudění vrátit

Označme chodby:

- **červená** prošel 2x (značky fixem)
- **žlutá** prošel 1x (natažená nit)
- **zelená** neprošel (ostatní)

Postup:

- Hledání začíná u Ariadny
- V každé místnosti postupuj odpředu podle tabulky:

1. je tu Minotaurus  $\Rightarrow$  STOP, našel
2. vede tu  $> 1$  žlutá  $\Rightarrow$  zpět, motej nit, **žlutá**  $\rightarrow$  **červená**
3. vede tu  $> 0$  zelená  $\Rightarrow$  vpřed, rozmotávej, **zelená**  $\rightarrow$  **žlutá**
4. je tu Ariadna  $\Rightarrow$  STOP, Minotaurus neexistuje
5. je tu 1 žlutá  $\Rightarrow$  zpět, motej nit, **žlutá**  $\rightarrow$  **červená**

# Další příklady na backtracking

## Problém dvou loupežníků

# Zrychlení backtrackingu ořezáváním

osm dam na šachovnici

- . přímo
- . sloupce
- . úhlopříčky

# Volba

dokážeme vrátit krok

**nebo**

musíme si pamatovat celý stav

# Zrychlení backtrackingu pomocí heuristik

## Heuristika

„dobrá rada“

„něco, co obvykle funguje“

loupežníci

proskákání šachovnice koněm



