13. Co víte o rekurzi a backtracingu?

# Rekurze

* Matematická: definování funkce nebo procesu pomocí jeho samého
* Programátorská: opakované vnořené volání stejné funkce
* Podmínky rekurzivního algoritmu:
  + Musí být definování podmínka pro ukončení algoritmu
  + V každém kroku rekurze musí dojít ke zjednodušení problému
  + V algoritmu se nejprve musí ověřit, zda nenastala koncová; pokud ne, provede se rekurzivní krok
* Typy rekurze:
  1. Přímá: Funkce volá sama sebe
  2. Nepřímá: f1() -> f2() -> f2() -> f1() ->…
* Pozitivní vlastnosti rekurze
  + Definovat nekonečnou množinu objektů konečným příkazem
  + Nekonečný počet výpočtů popsat pomocí konečného rekurzivního programu
  + Vytvořit poměrně krátké a efektivní programy
* Rekurzivní algoritmy mají velkou režii, protože každé volání podprogramu nutně způsobí:
  + Uložení lokálních proměnných na zásobníku
  + Předání parametrů a návratové adresy
  + Skok do podprogramu
  + Uvolnění lokálních proměnných
  + Návrat z podprogramu
* Kdy rekurzi nepoužít:
  + Máme-li k dispozici iterativní algoritmus se stejnou složitostí.
  + Chová-li se rekurzivní řešení nestabilně (např. se cyklus pro některá data zauzlí).
  + Pokud počet rekurzivních volání roste rychleji než lineárně

# Backtracking

* = prohledávání s návratem
* Úloha: Rozmístěte na šachovnici osm dam tak, aby se navzájem neohrožovaly
* Zkoušíme přípustná umístění dam, pokud nejde další dáma umístit, vrátíme se do předchozího stavu a zkusíme další možnost
* Tedy například: máme umístěné 4 dámy a všechna políčka na 5 řádku pro 5 dámu jsou ohrožována, vrátíme se a pro 4 dámu zvolíme další možné umístění