

- Dámy na šachovnici – dominance a nezávislost.
- Problémy řešitelné "vyplněním tabulky":
  - počet korektních uzávorkování pomocí  $n$  párů závorek,

# Dámy na šachovnici

- Jak rozmístit  $n$  dam na šachovnici  $n \times n$ , aby se vzájemně neohrožovaly?
- Jak rozmístit co největší počet dam, aby se navzájem neohrožovaly? (nezávislost)
- Jak rozmístit co nejmenší počet dam, aby ohrožovaly celou šachovnici? (dominance)

## $n$ dam na $n \times n$

- Každá dáma přijde do jednoho řádku,
- stačí dámu postupně zkoušet umísťovat na dosud neohrožená pole a spustit rekurzi na další řádky:

```
function dama(radek:integer);
begin
  if radek>pocet_radku then vypis
  else for i:=1 to pocet_radku do
        if volny_sloupec[i] and
volna_diag1??? and volna_diag2??? then
          begin
            obsad_to_vsechno;
            dama(radek+1);
            uvolni_to_vsechno;
          end;
        end;
end;
```

## Dámy na šachovnici

- Volné řádky můžeme evidovat v poli, abychom nemuseli prohledávat šachovnici.
- Volné diagonály taky, protože
- pro jednu diagonálu je součet konstantní ( $1 + 5 = 2 + 4 = 3 + 3 = 4 + 2 = 5 + 1$ ),
- pro druhou diagonálu je konstantní rozdíl ( $1 - 1 = 2 - 2 = 3 - 3 = \dots$ ),
- uděláme tedy tři pole booleanů.

## Dámy – dominance a nezávislost

- Situace je horší, nevíme předem, kolik jich bude,
- ale máme horní a dolní odhady. Pro dominanci i nezávislost nejvýše  $n$ , pro nezávislost stačí zkoušet každou dámu do jiného řádku.
- Kupříkladu napřed vždy zkus dámu nepřidat, pak až přidat. Tak ovšem nestačí jen najít první řešení, je potřeba projít všechny kandidáty (nikdo nezaručuje, že to s nejméně dámami najdeme první).

## Dámy – dominance a nezávislost

- Obecně funkční možnost: Zkusíme 1 –  $n$  dam přidávat na šachovnici všemi možnými způsoby a testovat, kolik polí ohrozíme, nebo zda ohrozíme jinou dámu.
- Kdo jaké triky vymyslí, takové má.
- Většinou nestačí pamatovat si, zda je pole ohrožené, ale kolikrát, abychom věděli, zda po odebrání "současné" dámy zůstane ohrožené nebo ne!

## Dominance a nezávislost – zobecnění

- Jiné šachovnice (tórus, Möbiův list, Kleinova láhev),
- jiné figurky (věž, kůň, Maharadža...),
- Více problémů spočívajících v horší analýze, kolik figurek stačí a kolik jich je naopak aspoň potřeba, lezení přes okraj pole...

## Počet korektních uzávorkování

- Jak budeme řešit?
- Pomocí rekurze podle rostoucího počtu přidaných závorek.
- Uděláme funkci, která:
  - zkusí přidat otevírací závorku (rekurze),
  - zkusí přidat zavírací závorku (rekurze),
  - pokud jsou použity všechny závorky, zvýš počet uzávorkování o 1.



## Počet korektních uzávorkování

```
var paru, celkem: longint;  
procedure pridej_zavorku(lev, prav: integer);  
begin  
    if lev > prav then  
        pridej_zavorku(lev, prav+1);  
    if paru > lev then  
        pridej_zavorku(lev+1, prav);  
    if (lev = prav) and (paru = lev) then  
        inc(celkem);  
end;
```