

## IV104 Dynamické programování

### Rostoucí podposloupnost

Napište program, který vypíše nejdelší rostoucí podposloupnosti zadané posloupnosti přirozených čísel (první číslo na vstupu je délka posloupnosti).

Příklad:

12 2 5 1 4 3 3 4 7 6 9 2 3

Výstup: 1 3 4 6 9.

### Nejdelší společná podposloupnost

Vstup: dva řetězce.

Výstup: nejdelší společná podposloupnost obou řetězců (pozor jde o podposloupnost nikoliv podřetězec, tj. nemusí to být souvislé).

Příklad: pro řetězce *pampeliska* a *sedmikraska* je výstup: *miska*.

### Hra Nim

Základní úloha: máme 23 sirek, v jednom tahu je povoleno odebrat 1, 3 nebo 4 sirky. Hráč, který nemůže táhnout (už nejsou žádné sirky), prohrál. Kdo má výherní strategii?

Obecná úloha: na vstupu máme počáteční počet sirek a seznam povolených tahů (kolik sirek je povoleno odebrat), výstupem je identifikace hráče, který má výherní strategii (a ideálně i výpis strategie v „rozumném formátu“).

### Hra s kulhavou dámou

Hra se hraje na čtvercové mřížce, na které jsou některá pole zablokována překážkou. Oba hráči mají jen jednu společnou figurku – kulhavou dámu, která se může pohybovat pouze směrem vlevo, nahoru nebo diagonálně vlevo-nahoru, v jednom tahu se může pohnout o libovolný počet polí ve zvoleném směru, nemůže však přeskakovat překážky. Hráči se střídají v tazích. Kdo nemůže táhnout, prohrává. Pole je pochopitelně omezené, takže nelze hrát mimo levý ani horní kraj.

Podle toho, jak jsou rozestaveny překážky a na jakém políčku dáma začíná, má buď první nebo druhý hráč výherní strategii (ať soupeř dělá cokoliv, hráč může vyhrát). Vaším úkolem je určit, kdo má kde výherní strategii.

Na webové stránce semináře máte ukázkové vstupy.

### Editační vzdálenost

Máme k dispozici následující tři editační operace: záměna znaku, smazání libovolného znaku, vložení znaku na libovolnou pozici. Na vstupu máme dva řetězce a úkolem je najít nejlevnější posloupnost editačních změn, která převede jeden řetězec na druhý.

Příklad: řetězce ostrov a saturn mají vzdálenost 5, ostrov  $\rightarrow$  strov  $\rightarrow$  satrov  $\rightarrow$  saturov  $\rightarrow$  saturnv  $\rightarrow$  saturn (2 smazání, 2 vložení, 1 náhrada).

### Maximalizace výrazu

Máme výraz bez závorek a máme najít uzávorkování tak, aby hodnota výrazu byla co nejvyšší. V jednodušší verzi uvažte pouze přirozená čísla a operace sčítání a násobení, v složitější verzi přidejte i odčítání a dělení.

Příklad: Pro vstup "2\*1+4\*1+3" dosáhneme maximální hodnoty pomocí uzávorkování "(2\*((1+4)\*(1+3)))".

Nemusíte řešit parsování vstupního řetězce – zadání vstupu udělejte tak, aby se vám s tím snadno pracovalo.