# 2. CVIČENÍ Z ADS1

pracujeme na poli i v matici

## Příklad první

Máme v poli $2n+1$ kladných celých čísel. Z toho je $n$ dvojic stejných a jedno, které dvojici nemá. Vymyslete algoritmus, který s co nejmenší pamětí v lineárním čase dokáže toto „nepárné“ číslo najít.

Pro zajímavost: jde to na jeden průchod polem a s jednou integerovou proměnnou.

## Příklad druhý

Mějme funkci $f(x)$ z přirozených čísel do přirozených čísel, jejíž definice je nám neznámá. Jediné, co o ní víme, je to, že posloupnost hodnot $0, f(0), f(f(0)), f(f(f(0))), \ldots$ se v nějakém kroku zacyklí. Vymyslete algoritmus, který se zastaví (to by jinak nebyl ani algoritmus) a vypíše délku cyklu posloupnosti popsané výše. Opět je třeba optimalizovat jak na čas, tak na paměť – řešení s lineární pamětí vás už není hodné.

Co se týče velikosti integerů, tak můžete předpokládat, že se každá hodnota $f(x)$ vejde do integeru a rovněž výsledná délka cyklu se do něj vejde.

## Příklad třetí

Mějme konvexní mnohoúhelník zadaný posloupností jeho vrcholů, jak leží za sebou na obvodu, proti směru hodinových ručiček, začínaje od vrcholu s nejnižší $x$-ovou souřadnicí. Vymyslete, jak rychle nalézt bod s nejvyšší $x$-ovou souřadnicí. Také vymyslete, jak najít bod s co nejnižší $y$-ovou souřadnicí.

## Příklad čtvrtý

*Oblíbená úloha Martina Mareše.* Mějme matici binárních čísel o velikosti $M \times N$ a hledáme v ní největší souvislou podmatici, která obsahuje jen samé nuly.

**Nápověda:** Vymyslete nejdříve algoritmus v $\mathcal{O}(N^2M^2)$, pak to zlepšete na $\mathcal{O}(NM^2)$ a pak přímo na $\mathcal{O}(MN)$. Zlepšování jde provádět pomocí předpočítávání nějakých počtů nul pro každý prvek matice.

Pokud na to jdete ale odjinud, tak nápovědu ignorujte a vymyslete to po svém – nikdy neexistuje univerzální nápověda.

## Příklad pátý

Prastarý je problém hledání *nejdelší rostoucí podposloupnosti* – máme pole $n$ přirozených čísel a hledáme největší možnou množinu indexů $i_1 < i_2 < i_3 \ldots$ tak, že $P[i_1] < P[i_2] < P[i_3] \ldots$. Nejprve vymyslete řešení v čase $\mathcal{O}(n^2)$.

## Příklad šestý

Nejdelší rostoucí podposloupnost jde vyřešit i v lepším čase, konkrétně v $\mathcal{O}(n \log n)$. Vymyslete, jak.

**Nápověda.** Když zpracovávám v posloupnosti další číslo, jde mi hlavně (ale nejen) o to, jestli se mi protáhne nejdelší rostoucí podposloupnost pomocí tohoto čísla nebo ne. Proč nejen? To nastiňuje příklad:

```
1 2 3 7 10 8 ...
```

I když osmička neprotahuje délku nejdelší rostoucí podposloupnosti, tak by měla asi naše pomocné údaje (pomocí kterých počítáme) nějak ovlivnit. To proto, že i když jedna nejdelší rostoucí podposloupnost končí desítkou, kdyby náhodou jako další číslo v posloupnosti následovala devítka, musíme vědět, že před devítku můžeme namísto desítky dát osmu. Všimněte si, že osmička prakticky desítku nahrazuje ...