

## Příklady o dokazování vlastností programů

Zkoušejte hledat řešení (důkazy) pro následující příklady. Jedná se tentokrát o docela obtížné věci, tak je klidně diskutujte s ostatními (a se cvičícími) na IS. (Doufám, že v textu nemám chyby...)

Mnoho zdaru!

**Příklad 2.1.** Představte si, že mezi  $N$  uzly jsou dány “sousední” vzdálenosti, přesněji řečeno, pokud  $i$  má přímé spojení s  $j$ , pak jeho délka je uložena v  $d[i][j]$  (jako dvourozměrné pole). Pokud  $i$  s  $j$  nemá přímé spojení, pak  $d[i][j] = \infty$ , tj. proměnná obsahuje “velmi velkou hodnotu”.

Potom následující krátký kód v C počítá matici nejkratších vzdáleností všech dvojic uzlů:

```
for (t=0; t<N; t++) {
    for (i=0; i<N; i++) for (j=0; j<N; j++)
        d[i][j] = min(d[i][j], d[i][t]+d[t][j]);
}
```

Uměli byste matematicky dokázat, proč tomu tak je?

(Návod: Zkuste indukci podle proměnné  $t$ . Jaké cesty jsou v algoritmu “započítány” pro konkrétní hodnoty  $t$ ?)

Byl by tento kód korektní, i kdyby vzdálenosti nebyly symetrické? (Tj. z  $a$  do  $b$  je třeba blíže než z  $b$  do  $a$ .)

Doplňkově se ptáme, co byste volili jako implementaci “velmi velké hodnoty”? Dejme tomu, že vzdálenosti jsou celočíselné (typu int), bylo by vhodné volit třeba konstantu MAX\_INT? (Proč ne?)

**Příklad 2.2.** Nechť dané pole  $p[]$  zapisuje permutaci  $n$  prvků  $0, 1, 2, \dots, n-1$ , tj. prvek  $i$  se v permutaci zobrazí na  $p[i]$ . (Permutace je bijekce množiny na sebe sama.) Zkuste se zamyslet, co potom počítá (jako hodnotu  $k$ ) následující kód v C:

```
k = 0;
for (i=0; i<n; i++) u[i] = 0;
for (i=0; i<n; i++) if (u[i]==0) {
    k++; u[i] = 1;
    for (j=p[i]; j!=i; j=p[j]) u[j] = 1;
}
```

(Umíte svou odpověď i dokázat? Proč vůbec by tento program měl ukončit běh?)