

Binární Vyhledávání

- z rozmezí indexů $1, \dots, n$ vrátí hledaný index, nebo nejbližší nižší (může vrátit 0 pokud vyhledávání vede řed první index)
- n - nejvyšší platný index
- $tooHigh(i)$ - otestuje index, zda je vyšší než hledaná hodnota

Function $BinSearch(n, funkce\ tooHigh)$

```
 $L \leftarrow 0$ 
 $R \leftarrow n + 1$ 
if  $n = 0$  then
  | return "no-valid indices"
end
while  $L + 1 < R$  do
  |  $M \leftarrow \lfloor \frac{L+R}{2} \rfloor$ 
  | if  $tooHigh(M)$  then
  | |  $R \leftarrow M$ 
  | end
  | else
  | |  $L \leftarrow M$ 
  | end
end
return  $L$ 
end
```

Invarianty

- vždy platí $L < M < R$ po výpočtu M . Lze díky podmínce cyklu vždy platí $L + 1 < R$, invariant plyne přímo. Speciálně platí $L < R$ před i po změně hranic.
- pro hledaný index X uvnitř cyklu vždy platí $L \leq X < R$. Na počátku výpočtu jistě platí. R se mění pouze na hodnoty větší než X a L se mění pouze na hodnoty, které nejsou větší než X

Konečnost

V každém cyklu se jedna z hranic nastaví na M . Z prvního invariantu plyne, že se tato hranice změní a to tak, že se L a R přiblíží.

Korektnost

Vyjma případu pro $n = 0$ algoritmus vstoupí do cyklu. Po ukončení cyklu z druhého invariantu máme $L \leq X < R$ (pro X hledanou hodnotu). Z podmínky ukončení cyklu $L + 1 \geq R$, z prvního invariantu $L < R$ a tedy $L + 1 = R$. Dohromady $L \leq X < L + 1 = R$. Pokud X je celočíselný index, potom je roven L , pokud X je mezi indexy, potom je L nejbližší nižší index.
