



Algoritmy a datové struktury 1

[Back to the course](#)

DÚ1: první díra

Deadline: 2022-03-03 10:40 (60 days ago)

Pavel Veselý — 2022-02-17 17:55 (74 days ago) — [reply](#)

Na vstupu dostaneme rostoucí posloupnost přirozených čísel. Chceme najít nejmenší přirozené číslo, které v ní chybí.

Pavel Veselý — 2022-02-17 21:52 (73 days ago) — [reply](#)

Upřesnění zadání: posloupnost je konečná a známe její délku n . Chceme dosáhnout asymptoticky lepší než lineární časové složitosti (za ní nebudou body).

Zdeněk Tomis — modified 2022-02-18 16:54 (73 days ago) — [edit](#) — [reply](#)

Algoritmus

Použijeme obdobu binárního vyhledávání.

Pokaždé se podíváme na prostřední prvek (nebo jeden z prostředních) z uvažované posloupnosti pokud jeho hodnota odpovídá jeho indexu, víme, že díru v první části nenajdeme a musíme hledat napravo v posloupnosti (podle prostředního prvku). V opačném případě, když je hodnota větší než index, víme, že se díra nachází v části před vybraným prvkem.

Pseudokód

Všechna indexování v pseudokódu začínají od nuly

funkce najdi díru (posloupnost)

konec zkoumané podposloupnosti = konec = délka posloupnosti - 1

začátek zkoumané podposloupnosti = začátek = 0

dokud délka zkoumané posloupnosti > 1

```

zkoumaný index = floor( (konec - začátek) / 2) + začátek

pokud posloupnost[zkoumaný index] != zkoumaný index + 1

    konec = zkoumaný index

jinak

    začátek = zkoumaný index + 1

pokud posloupnost[začátek] != začátek + 1

    vrať začátek + 1

// tento případ nastane pouze tehdy, když je posloupnost bez díry a je
// nutno vrátit člen nad maximem

vrať začátek + 2

```

Správnost

Algoritmus v každé iteraci určí, ve které části se díra vyskytuje. Invariantem je fakt, že díra se určitě nachází ve “zkoumaném úseku” nebo těsně za ním, jelikož při prozkoumání zvoleného pivotu je nutně pravda, že pokud je hodnota pivotu odpovídá jeho indexu (například na páté pozici je pětka), musí být podposloupnost před pivotem bez díry a díra se tedy nachází buď v zbylém úseku, nebo hned za ním, pokud je posloupnost sama o sobě bez díry.

Pojem díra zde používám jako první index, na kterém není odpovídající hodnota, což je trochu rozdíl oproti zadání, kde se jedná o tu samotnou hodnotu, která se tedy nikde nenachází. Toto pojmenování se ale hodí.

Konečnost

Zkoumaný úsek se vždy zmenší alespoň o jedna, když bude velikosti jedna, nastane finální krok algoritmu.

Časová složitost

$O(\log n)$

V algoritmu se vyskytuje cyklus, který ale proběhne nejvýše $\log_2 n$, a to proto, že se v každém kroku zmenší zkoumaný úsek na jednu polovinu, nastane totiž jedna ze dvou možností:

- index odpovídá očekávané hodnotě, pak se posune začátek a zavrhneme první polovinu úseku
- index neodpovídá očekávané hodnotě, pak se posune konec a první díru hledáme už v první polovině zrovna zkoumaného úsek

Platí pak $\text{počet iterací} = \log_2 n$

Je pravda, že nově vzniklý úsek může mít o jednotku větší délku než je polovina, asymptotickou složitost to ale neovlivní.

Prostorová složitost

$O(1)$

Algoritmus zřejmě využívá pouze konstantní paměť a paměť, ve které je načten vstup.

Pavel Veselý — 2022-02-21 18:13 (70 days ago) — [reply](#)

Výborně, pěkně sepsáno.

Points: 8.00

[Return](#)

New post (You can use [Markdown](#) with [KaTeX math](#) here)

Attachment (PDF or UTF-8 text): Keine Datei ausgewählt.

[Submit](#)

[Preview](#)

The Owl is maintained by [Martin Mareš](#). Send all suggestions, bug reports, and requests for new courses to mj@ucw.cz.