Hledání prvočísel, Eratosthenovo síto

# Úvodní informace

Tato úloha vychází z populární a intuitivní metody hledání všech prvočísel v poli o určité velikosti. Studenti nejprve vymýšlí vlastní metody, jakými by dosáhli výčtu všech prvočísel v zadaném limitu. Poté jsou navedeni na princip Eratosthenovo síta a jeho následnou implementaci.

# Cíle úlohy

* Procvičení:
  + Cykly
  + Vnořené cykly
  + Pole
* Podněcování skupinového brainstormingu

# Náročnost

* 2 vyučovací hodiny
* Obtížnost:

# Prerekvizity

* Cykly a vnořené cykly
* Pole
* Definice prvočísla (je případně součástí úvodního brainstormingu)
* Matematická knihovna (funkce pro práci s odmocninou), případně ji lze představit v průběhu výuky

# Metodika výuky

Tato úloha je rozdělena na dvě části. V první fázi jsou studenti postaveni před problém – najdi všechna prvočísla menší než N a následnému vylepšení programu. V druhé jde o programovou realizaci síta.

Zadání je pro obě části stejné a může znít takto:

Najděte nejefektivnější algoritmus pro hledání prvočísel v zadaném limitu. Pro práci s číselným rozsahem využijte prvků indexovaného pole, kde jednotlivé klíče pole budou reprezentovat celočíselnou testovanou posloupnost a hodnoty pole budou určovat prvočíselnost oněch klíčů.

Vstup: 100

Výstup: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

## Dedukce – hledání hrubou silou

Pomocí brainstormingu zde můžeme hned na začátku přijít na definici prvočísla. Položme tedy studentům otázku: *Co je to prvočíslo?*

Je pravděpodobné, že v tomto případě dojdeme rychle ke správné odpovědi, jelikož se jedná o známou definici. Pro kontrolu ještě můžeme přidat další otázku: *Kdo si myslí, že číslo 80 je prvočíslo?*

Kdo se přihlásí, zjevně nepochopil definici. Je proto nutné vysvětlit, proč tomu tak není. Pokračujme v otázkách i dále: *Jak spolehlivě zjistíme, že nějaké číslo je prvočíslo?*

Někdo se vytasí s poučkami o ciferných součtech, sudosti a dalších poučkách pro jednotlivé dělitele, ale ty nepokryjí všechny případy. Je třeba diskusi směřovat k závěru, že se mně číslo N nepodaří vydělit beze zbytku čísly od 2 do N-1. Možná někdo už přijde s odmocninou (viz níže). Abychom to ale nasměrovali žádoucím směrem, můžeme vyslovit požadavek, že chceme nejjednodušší algoritmus z hlediska zápisu, resp. náročnosti na pochopení, takže návrh na odmocninu a podobné zlepšováky oceníme, ale prozatím vynecháme.

Po brainstormingu definice prvočísla pracují studenti samostatně na svých programech. Pravděpodobným výstupem této samostatné práce jsou řešení hrubou silou (více, či méně efektivní).

Efektivita se odvíjí od způsobu, jakým je při zjišťování testována dělitelnost beze zbytku:

* Všechna menší čísla než je zjišťované číslo N – velká zátěž
* Všechna menší čísla než je N/2 – velká až střední zátěž
* Všechna menší čísla než je odmocnina z N – malá zátěž

Další důvody negativního ovlivnění rychlosti:

* Při nalezení dělitele není cyklus ukončen a nadále testuje zbytek možných dělitelů
* Možný dělitel je testován od největšího čísla po nejmenší

V dalším kroku je vhodné promítnout přiložený program 1\_brute-force v preferovaném jazyce a provést diskusi. Pro začátek mohou studenti hledat odlišnosti v jejich a promítaném programu (ve skupinách, či dohromady). Algoritmus je velice málo efektivní a zahrnuje všechny nedostatky shrnuté výše. Je pravděpodobné, že studenti nedokáží přijít na minimální počet testovaných čísel, tedy odmocnina z čísla N.

Pokud přišel někdo z nich na limit N/2, položme otázku: *Můžeme jít ještě níž?*

V případě, že všichni testovali až do čísla N-1: *Je nutné testovat všechna čísla?*

Když žádná z těchto otázek nepodpoří navedení na správnou cestu, nic se neděje a prostě jim to vysvětlíme. Jestliže postupuji od 2 dále a poprvé se mi podaří N vydělit nějakým A, pak N / A má celočíselný výsledek, který označíme B, a toto B je logicky větší než A (kdyby nebylo, našli bychom ho dříve než A) a je to taky dělitel, jež je logicky menší než N. Odmocnina je pak jasná hranice, protože čím větší je A, tím menší je B, ale stále je A < B a tou odmocninou najdeme hranici, kdy A = B (i když jen teoretickou – často nevyjde celočíselně). K odbornému vysvětlení je možné provést důkaz:

Nechť N je součin libovolných přirozených čísel A a B, která jsou větší než 1 a zároveň menší než N:

Výraz A je větší nebo rovno B vynásobíme A a B, vzniknou dvě rovnice:

Následně můžeme nerovnice sloučit a dosadit N:

Jelikož počítáme v množině přirozených čísel, můžeme výsledný výraz odmocnit, z čehož vyplívá, že není možné, aby obě čísla byla zároveň větší, než je odmocnina z N:

Nemělo by se ale zapomenout i na vysvětlení ukončení cyklu a proč je lepší testovat dělitelnost od dvojky výše. Takovou otázku můžeme položit opět studentům, protože odpověď je triviální. *Jaká je pravděpodobnost, že číslo vydělím 2?* 50 %. *Jaká je pravděpodobnost, že číslo vydělím 3?* 33 %.

Následně by měli studenti upravit svůj program podle diskutovaných výkonových zlepšení. V případě nutnosti ukážeme způsob, jak se ve zvoleném programovacím jazyce implementuje matematická knihovna a práci s ní.

### Otázky do diskuse

1. Najdete rozdíly mezi vaším a promítaným programem?
2. Mají tyto odchylky vliv na celkový výkon programu?

### Možné problémy

* Student nedokáže implementovat algoritmus definice prvočísla*.* – Explicitní navedení na libovolném prvočísle pomocí důkazu. Proč je číslo 7 prvočíslo?
* Student neví, jak do programu zahrnout čísla 0 a 1. – Vysvětlit všem, že se jedná o ukázku toho, že nastanou situace, kdy není možné algoritmus použít na všechnu množinu čísel a je nutné některé prvky upravit (v našem případě vyjmout) „natvrdo“.

## Eratosthenovo síto

Po předchozí diskusi představíme Eratosthenovo síto. Je dobré použít přiložený soubor eratosthenovo-sito.gif, který znovu poslouží k dedukci a následné debatě nad způsobem fungování síta.

Princip Eratosthenovo síta je následující: Pro čísla z rozsahu 2 až N vezmeme nejmenší číslo (v prvním kroku 2). Toto číslo je zároveň prvočíslem. Odstraníme (v pomocném slidu vyškrtneme) všechny jeho násobky v rozsahu. Poté pokračujeme dalším nevyškrtnutým číslem (v druhém kroku 3) a provedeme stejnou operaci. Takto pokračujeme až do doby, kdy je další číslo větší než odmocnina z N (důvod byl popsán v první části), všechna zbylá čísla jsou prvočísla.

Toto je možné vysvětlit také pomocí přiloženého souboru: eratosthenovo-sito.png.

Obsah obrázku text

Popis byl vytvořen automaticky

Obrázek - Vysvětlení Eratosthenovo síta

Studenti mají za úkol samostatně implementovat Eratosthenovo síto. Pro rychlejší práci mohou použít již napsaný program z první fáze a pouze v něm upravit cykly, které určují prvočíselnost. Výsledný program je přiložen jako 2\_eratosthenovo-sito.

Pro zpestření výuky je součástí také spustitelný soubor 3\_porovnani-vykonu.exe (samotný kód programu se nachází v souboru 3\_porovnani-vykonu.cpp). Program testuje rychlost použitých programů pomocí knihovny chrono (bez vypisování prvočísel).

### Otázky do diskuse

1. Jakým způsobem funguje Eratosthenovo síto?
2. Jaká je horní mez zkoumaných čísel?
3. Kolikrát rychlejší bude Eratosthenovo síto, než program promítaný v první fázi?

### Možné problémy

* Se znalostmi z první části se nepředpokládají dílčí problémy.