Lineární vyhledávání, binární vyhledávání

# Úvodní informace

V této úloze budou studenti pomocí lineární metody a vyhledávání půlením intervalu zjišťovat, zda se hledaná hodnota nachází v poli. V případě úspěšného nalezení prvku dojde k vypsání indexu, na kterém se hodnota nachází, v opačném situaci vypíšeme hlášku, že zadaná hodnota se nenachází v poli. V diskusi debatujeme nad tím, co je to rozděl a panuj metoda a její využití v praxi.

# Cíle úlohy

* Procvičení:
  + Cykly
  + Rekurze
  + Funkce vlastní
* Diskuse nad metodou rozděl a panuj
* Zefektivnění algoritmu pomocí různého přístupu k řešení

# Náročnost

* 2-3 vyučovací hodiny
* Obtížnost:

# Prerekvizity

* Cykly
* Rekurze
* Funkce vlastní

# Metodika výuky

Úlohu si rozdělíme na dva samostatné příklady. V prvním dostanou studenti za úkol bez jakýchkoli dodatečných informací zjistit, jestli se hledaná hodnota nachází v seřazeném poli. V reflexi prvního příkladu představíme studentům binární vyhledávání, které si studenti v následně vyzkoušejí naprogramovat.

## Lineární vyhledávání

Zadání:

Napište program, který se pokusí v seřazeném poli nalézt index, na kterém se bude nacházet hledaná (vstupní) hodnota. V případě nalezení program vypíše, na jakém klíči se hodnota nachází. V opačné situaci vypíše hlášku, že se hodnota v poli nenachází. Hodnoty pole mohou být pouze přirozená čísla. Pro hledání použijte funkci linearneNajdi, která v případě úspěšného nalezení vrátí index hledané hodnoty, v opačné situaci vrátí -1.

Pole: [10, 20, 30, 35, 40, 45, 50, 70, 80, 120, 140, 141, 150, 160]

Vstup: 141

Výstup: Hodnota 141 se nachazi v poli na indexu 11.

Vstup: 130

Výstup: Hodnota 130 se nenachazi v poli.

Jedná se o velice triviální úlohu, ke které není potřeba dělat žádný rozbor. Po představení zadání se studentů zeptáme, jestli úloze všichni rozumí a ví, jak ji řešit. V případě, že někdo neví, můžeme nechat ostatní vysvětlit, jakým způsobem budeme postupovat.

Studenti budou procházet pole prvek po prvku a testovat, jestli je shodný s hledaným. Předpokládá se, že studenti ví, jak v programovacím jazyce zjistí velikost pole (kolik prvků se v něm nalézá). Způsob zjištění je v každém odlišný a v případě, že jej neznají, je dobré jim ho představit.

Vzorové řešení se nalézá v 1\_linearni-vyhledavani. Po úspěšném vyřešení příkladů můžeme navázat se studenty diskusi.

### Otázky do diskuse

1. Způsob, který jsme využili, se jmenuje lineární vyhledávání. Přijde vám v něčem nevýhodné nebo neefektivní?
2. Porovnejte efektivitu algoritmu ve vzorovém poli na hodnotách 20 a 150.

Výstupem z diskuse by mělo být zjištění, že zvláště pro hodnoty nacházející se ke konci pole (na horních indexech) je hledání pomalé. Můžeme se jich zeptat, jak by toto zlepšili? Je možné, že některý z nich vymyslí testování každého druhého čísla. V případě, že bude hledané číslo menší než hodnota testovaného indexu, tak otestují to vynechané. Pokud tohle nevymyslí, tak jim tento způsob navrhneme a necháme o něm opět diskutovat a budeme pouze moderovat, jakým směrem se bude rozprava ubírat*. Proč netestovat každé třetí, čtvrté, páté .. dvanácté číslo? Od čeho se odvíjí, jaký index testovat, abychom dosáhli nejefektivnějšího výsledku?* Toto mohou být otázky, pomocí kterých se budeme snažit studenty navést na princip binárního testování, ke kterému plynule přejdeme.

### Možné problémy

* Díky triviálnosti úlohy se neočekávají žádné problémy.

## Binární vyhledávání

Pro vysvětlení algoritmu binárního vyhledávání je možné použít přiložený soubor binarni-vyhledavani.png. Je možné takový rozbor provést na tabuli na jiných číslech. Princip tohoto algoritmu spočívá v rozdělení si intervalu vždy na polovinu. Testujeme, jestli je hodnota nacházející se uprostřed pole větší, menší, nebo rovna hledanému číslu. V případě rovnosti vrátíme index. V situaci, kdy je větší, nebo menší, opakujeme část rozdělení intervalu na polovinu jen s tím rozdílem, že interval bude jen v daném úseku, kde by se hledané číslo mělo vyskytovat.

Obsah obrázku stůl

Popis byl vytvořen automaticky

Obrázek - Pomocný obrázek k vysvětlení binárního vyhledávání

Z pomocného obrázku může vzniknout rozpačitý výsledek. *Vždyť jsme testovali nakonec stejný počet indexů o dost složitěji a nedosáhli jsme žádné časové (výpočetní) úspory.* V tomto konkrétním případě budou mít studenti pravdu. Důvodem nastavení hledané hodnoty právě na 3 v tomto konkrétním poli je podrobné vysvětlení všech variant, které by mohly nastat. V situaci, kdy se do takového stavu dostaneme, tak jej vyzveme, aby vyzkoušel číslo 25 nebo 66 na stejném poli. Tam už je situace odlišná, pro 25 je počet testovaných hodnot roven dvěma a pro 66 třem. Tady se ukazuje síla binárního vyhledávání, která nemusí být na první pohled všem zřejmá.

Nyní se již můžeme pustit do samotného řešení.

Zadání:

Napište program, který se pomocí binárního vyhledávání pokusí v seřazeném poli nalézt klíč o hledané (vstupní) hodnotě. V případě nalezení program vypíše, na jakém klíči se hodnota nachází. V opačné situaci vypíše hlášku, že se v poli nenachází. Za hodnoty pole považujte pouze přirozená čísla. Pro hledání použijte funkci binarneNajdi, která v případě úspěšného nalezení vrátí index, v opačné situaci vrátí -1.

Pole: [10, 20, 30, 35, 40, 45, 50, 70, 80, 120, 140, 141, 150, 160]

Vstup: 141

Výstup: Hodnota 141 se nachazi v poli na indexu 11.

Vstup: 130

Výstup: Hodnota 130 se nenachazi v poli.

Příklad má dva způsoby přístupu k řešení. Jde o rekurzivní a iterativní. Záleží tedy jen na nás a časové dotaci, který přístup (případně oba) zvolíme. Je málo pravděpodobné, že studenti dokáží vymyslet rekurzivní způsob, i tak je vhodné jej nakonec představit a vysvětlit. Jednotlivá řešení jsou uložena pod zdrojovými kódy 2\_binarni-vyhledavani-rekurze pro rekurzivní přístup a 2\_binarni-vyhledavani-cyklus pro iterativní přístup.

Po úspěšném vyřešení je možné pokračovat v diskusi nad tímto příkladem. Vhodné je v ní vysvětlit a pozastavit se nad problematikou metody rozděl a panuj, kterou jsme ve způsobu řešení nevědomky využili.

### Otázky do diskuse

1. Přijde Vám binární vyhledávání efektivnější než lineární?
2. Jaké jsou nevýhody této metody oproti lineárnímu hledání? Pro které indexy může být lineární vyhledávání efektivnější?
3. Využili jsme zde metodu rozděl a panuj. Najděte na internetu, o co se jedná a kde je přesně v našem případě aplikována.
4. Co je to camelCase notace (vyhledejte na internetu)? Použili jsme ji někde v tomto příkladu?
5. Jaké další alternativy značení funkcí a proměnných jsou používány? (vyhledejte na internetu)

### Možné problémy

* I přes relativní obtížnost algoritmu po důkladném vysvětlení nejsou očekávány žádné problémy. Důležité je se ujistit, jestli studenti porozuměli binárnímu vyhledávání.