5. Vyhledávání pozice prvku za použití algoritmu Binary search

David Mrlík
10.12.2022

Obsah:

1. Zadání semestrální práce

Zapište program, který do vzestupně setříděné posloupnosti kladných čísel bude zařazovat postupně další prvky a to pouze v případě, že se příslušný prvek ve vytvářené posloupnosti doposud nevyskytuje. Při vyhledání pozice pro vložení dalšího prvku využijte setříděnosti posloupnosti – aplikujte binární vyhledávání.
Specifikace vstupu
Program má umožnit při jednom spuštění zpracování libovolného počtu zadání. Před zadáním další úlohy nechť program vypíše dotaz uživateli, zda pokračovat ve zpracování či nikoli – odpověď uživatele bude znak 'a' nebo 'n' (malými nebo velkými písmeny). Program má skončit v případě, že odpověď uživatele je 'n'. Při načítání počáteční posloupnosti nechť program nejdříve načte počet a poté jednotlivé hodnoty posloupnosti. Poté má program načítat další čísla do zadání záporné hodnoty a tato čísla postupně zařazovat do posloupnosti. Načtením záporného čísla zpracování aktuální úlohy končí, program má vypsat výslednou posloupnost.

2. Pojetí úlohy

2.1. Nejasnosti

V úloze není specifikováno, co se má dít s posloupností, pokud není vzestupně setříděná. První verze mého programu zadanou posloupnost nejprve setřídila pomocí algoritmu Bubble sort, a poté s ní pracovala jako podle postupu uvedeného výše. Dále není specifikováno, co se má dít s posloupností, která obsahuje záporná čísla. A také není ze zadání zřejmé, zdali zadaná posloupnost má mít všechny prvky unikátní (vyskytující se nejvýše jednou). Všechny tyto podmínky jsou ošetřeny.
Finální verze programu vstupní posloupnost netřídí, ale ověřuje její setříděnost pomocí postupu, který je popsán v bodě 2 v sekci návrh řešení.

2.2 Předpoklady

Program je funkční za předpokladu dodržení podmínek uvedených výše a za předpokladu, že uživatel bude v průběhu programu zadávat pouze celá čísla (vyhovující podmínkám v zadání). Dále je nutné dodržet podmínku o velikosti zadané posloupnosti, která musí být větší než 0. Pokud je velikost zadané posloupnosti záporná či rovna nule, dojde k vypsání chybové zprávy na konzoli a program začne požadovat novou velikost posloupnosti. Také je nutné dodržet podmínku o setříděnosti posloupnosti. Pokud je zadaná posloupnost nesetříděná, dojde k vypsání chybové zprávy na konzoli a program začne požadovat novou velikost posloupnosti. A naposled je nutné dodržet podmínku o unikátnosti posloupnosti. Pokud se v zadané posloupnosti vyskytuje duplikát některého z již předem zadaných prvků, dojde k vypsání chybové zprávy na konzoli a program začne požadovat novou velikost posloupnosti.

3. Návrh řešení:

1. Načtu vstupní posloupnost
2. Ověřím, zda je posloupnost vzestupně setříděná.
   2.1. Pomocí for cyklu, který půjde od jedničky do posledního prvku v poli.
   2.2. Budu porovnávat prvek pole s jeho předchozím. Pokud bude předchozí prvek větší nebo roven aktuálnímu prvku, tak zadaná posloupnost není vzestupně setříděná.
3. Začnu načítat čísla, které budu dosazovat na správné místo v posloupnosti
4. Ověřím zdali se zadaný prvek vyskytuje v posloupnosit. Pokud ne program dále běží. Pokud ano zobrazím zprávu v konzoli, že se zadaný prvek v posloupnosti již vyskytuje.
5. Abych mohl dosadit čísla do setříděné posloupnosti, budu si muset pro každé nově přidané číslo založit nové pole, do kterého dám hodnoty z předchozího pole a jeho velikost zvětším o jedna
6. Zadaný prvek zatřídím do setříděné posloupnosti pomocí algoritmu Binary Search
   6.1. Na vstupu dostane funkce rozšířené pole, prvek, který chceme zatřídit, levou a pravou stranu. Levá strana představuje index prvního prvku posloupnosti a pravá strana představuje index posledního prvku v posloupnosti (nezvětšeného pole).
   6.2. Vypočítám střed posloupnosti.
   6.3. Nejprve porovnám zadaný prvek s největším prvkem v posloupnosti (uložený na indexu right). Pokud je větší, pak pozice zadaného prvku je poslední index v novém zvětšeném poli. Následně ho porovnám s prvním prvkem v poli (uložený na indexu left). Pokud je menší, pak je jeho pozice v novém poli uložena na indexu 0. Pokud prvek nesplňuje ani jednu z výše uvedených podmínek, pak aplikuji algoritmus binary search
   6.4. Porovnávám zadaný prvek se středem. Pokud je větší, hledám v pravé části pole, která začíná středem + 1 a končí posledním prvkem v posloupnosti. Pokud je menší, hledám v levé části pole, která začíná prvním prvkem v posloupnosti a končí bodem střed -1. Vložím do něj prvky z původního zvětšeného pole a vracím toto zmenšené pole. (Abych se vyhnul kopírováním nuly na posledním místě zvětšeného pole).
   6.5. Takto dále dělím pole, dokud nenaleznu prvek v posloupnosti nebo dokud se nedostanu do bodu, kdy je index levé strany rozděleného pole větší nebo roven pravé. V tom případě porovnávám prvek s prvkem na indexu levé strany. Pokud je daný prvek menší ukládám do jeho pozice index levé strany. Jinak do jeho pozice ukládám index o jeden větší než index prvku na levé straně.
   6.6. Důvodem indexování do levé strany je výpočet středu. V momentě, kdy levá strana + pravá, není dělitelná dvěma, přičítám k tomuto výsledku 1 a až poté dělím dvěma, abych získal střed. To zapříčiní, že v momentě, kdy je levá strana o jedna menší než pravá, rovnost mezi proměnnými střed a pravá. A pokud je prvek na středu větší než zadaný prvek, zvětší se hodnota levé strany. A tím, že hodnota pravé strany byla rovna středu a vím, že je zadaný prvek větší než střed, tak do něj ukládám hodnotu z levé strany.

7. Nyní znám pozici prvku v posloupnosti. K tomu abych mohl prvek zapsat na místo posloupnosti a zároveň si tím nepřepsal hodnotu prvku, který se na tom místě momentálně nachází, tak musím posunout prvky pole, které jsou na vyšší pozici než je daný prvek.

8. Toho docílím pomocí for cyklu, který jde od posledního místa v novém poli (je tam uložena hodnota 0) do pozice mého prvku, přičemž do nové pozice ukládám prvek, který byl předtím o pozici níže. Tím zajistím, že prvek, který by byl přepsán zadaným prvkem, bude zduplikován. Následně zapíšu zadaný prvek na jeho pozici a vracím zpátky pole se zatříděným zadaným prvkem.

4. Protokol z testování

Číslo testu	**Typ testu, popis vstupů**	**Očekávaný výsledek**	**Skutečný výsledek**	**Prošel (ano/ne)**
1.	Posloupnost vyhovující všem podmínkám **(1, 12, 30, 44, 61)** Testuji pro hodnoty vyhovující všem podmínkám **(5, 10, 7, 8)**	**1, 5, 7, 8, 10, 12, 30, 44, 61**	**1 5 7 8 10 12 30 44 61**	**ano**
2.	Posloupnost nevyhovující podmínce o setříděnosti **(3, 15, 7, 14)**	Oznámení v konzoli o zadání nesetříděné posloupnosti a vyzvání k zadání posloupnosti nové	Zadaná posloupnost není setříděná ! Zadej počet prvků počáteční posloupnosti:	**ano**
3.	Posloupnost vyhovující všem podmínkám **(15, 16, 17, 18)** Testuji pro hodnoty vyhovující všem podmínkám **(1, 2, 3, 20)**	**1, 2, 3, 15, 16, 17, 18, 20**	**1 2 3 15 16 17 18 20**	**ano**
4.	Posloupnost nevyhovující podmínce o nulovém či záporném počtu. **(-4)**	Oznámení v konzoli o zadání záporného či nulového počtu prvků v posloupnosti.	Zadaná posloupnost nemůže být délky 0 či záporná ! Zadej počet prvků počáteční posloupnosti:	**ano**
5.	Posloupnost vyhovující všem podmínkám **(4, 17, 100, 900)** Testuji pro hodnoty nevyhovující podmínce o unikátním výskytu prvku v posloupnosti **(4, 90, 81, 90)**	Pro prvky **4** a **90** (při druhém zadání) Oznámení v konzoli o výskytu prvku v posloupnosti a vyzvání k dalšímu zadávání Výsledná posloupnost **(4, 17, 81, 90, 100, 900)**	Zadej počáteční vzestupně setříděnou posloupnost: **4 17 100 900** Zadávej další hodnoty: **4** Zadaný prvek se v posloupnosti již vyskytuje ! Zadávej další hodnoty: **90** Zadávej další hodnoty: **81** Zadávej další hodnoty: **90** Zadaný prvek se v posloupnosti již vyskytuje ! Zadávej další hodnoty: **-1** **4 17 81 90 100 900**	**ano**

5. Screenshoty výsledků z akceptačních testů:

```
Zadej počet prvků počáteční posloupnosti:
5
Zadej počáteční vzestupně setříděnou posloupnost:
1 12 30 44 61
Zadávej další hodnoty:
5
Zadávej další hodnoty:
10
Zadávej další hodnoty:
7
Zadávej další hodnoty:
8
Zadávej další hodnoty:
-1
1 5 7 8 10 12 30 44 61
Pokračovat ve zpracování (a/n):a
Zadej počet prvků počáteční posloupnosti:
4
Zadej počáteční vzestupně setříděnou posloupnost:
3
15
7
14
Zadaná posloupnost není setříděná !
Zadej počet prvků počáteční posloupnosti:
4
Zadej počáteční vzestupně setříděnou posloupnost:
15 16 17 18
Zadávej další hodnoty:
1
Zadávej další hodnoty:
2
Zadávej další hodnoty:
3
Zadávej další hodnoty:
20
Zadávej další hodnoty:
-1
1 2 3 15 16 17 18 20
Pokračovat ve zpracování (a/n):a
```

```
Zadej počet prvků počáteční posloupnosti:
-4
Zadaná posloupnost nemůže být délky 0 či záporná !
Zadej počet prvků počáteční posloupnosti:
4
Zadej počáteční vzestupně setříděnou posloupnost:
4 17 100 900
Zadávej další hodnoty:
4
Zadaný prvek se v posloupnosti již vyskytuje !
Zadávej další hodnoty:
90
Zadávej další hodnoty:
81
Zadávej další hodnoty:
90
Zadaný prvek se v posloupnosti již vyskytuje !
Zadávej další hodnoty:
-1
4 17 81 90 100 900
Pokračovat ve zpracování (a/n):n
davidm@davidm-ThinkPad-T550:~/study/ALG1/SemestralniPrace$
```

5.1. Extra test pro zadání posloupnosti délky 50 a pro počet prvků
Zadání pomocí třídy test, která bere náhodné hodnoty

```
Zadaná posloupnost:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 42 44 51 51 52 54 55 62 63 67 69 72 72
73 77 81 81 82 82 83 83 89 92 92 93 95 96 97 98

Prvek k setřídění: 38

Pole po zadání prvku k zatřídění:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 62 63 67 69 72
72 73 77 81 81 82 82 83 83 89 92 92 93 95 96 97 98

Prvek k setřídění: 58

Pole po zadání prvku k zatřídění:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 89 92 92 93 95 96 97 98

Prvek k setřídění: 36

Pole po zadání prvku k zatřídění:
Zadaný prvek se v posloupnosti již vyskytuje !
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 89 92 92 93 95 96 97 98

Prvek k setřídění: 67

Pole po zadání prvku k zatřídění:
Zadaný prvek se v posloupnosti již vyskytuje !
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 89 92 92 93 95 96 97 98

Prvek k setřídění: 90

Pole po zadání prvku k zatřídění:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 89 90 92 92 93 95 96 97 98
```

```
Prvek k setřídění: 44

Pole po zadání prvku k zatřídění:
Zadaný prvek se v posloupnosti již vyskytuje !
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 89 90 92 92 93 95 96 97 98

Prvek k setřídění: 84

Pole po zadání prvku k zatřídění:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 51 51 52 54 55 58 62 63 67 69
72 72 73 77 81 81 82 82 83 83 84 89 90 92 92 93 95 96 97 98

Prvek k setřídění: 50

Pole po zadání prvku k zatřídění:
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 50 51 51 52 54 55 58 62 63 67
69 72 72 73 77 81 81 82 82 83 83 84 89 90 92 92 93 95 96 97 98

Prvek k setřídění: 5

Pole po zadání prvku k zatřídění:
Zadaný prvek se v posloupnosti již vyskytuje !
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 50 51 51 52 54 55 58 62 63 67
69 72 72 73 77 81 81 82 82 83 83 84 89 90 92 92 93 95 96 97 98

Prvek k setřídění: 92

Pole po zadání prvku k zatřídění:
Zadaný prvek se v posloupnosti již vyskytuje !
2 5 6 7 8 14 20 21 23 23 24 27 29 29 29 33 33 33 34 36 36 38 42 44 50 51 51 52 54 55 58 62 63 67
69 72 72 73 77 81 81 82 82 83 83 84 89 90 92 92 93 95 96 97 98
```