

Domácí úkol k cvičení číslo 5

29. března 2024

1 Příklady ke složitosti

Doporučuji postupovat podle přednášky, případně podle knihy „Introduction to the Design and Analysis of Algorithms.“

1.1 Vyřešte rekurentní rovnici

Najděte funkci T takovou, že $T(0) = 1$ a pro všechna $n \in \mathbb{N}$ (všechna kladná celá čísla) platí¹

$$T(n) = 5T(n - 1). \quad (1)$$

Prosím stručně okomentovat postup řešení.

2 Úloha k naprogramování

Naprogramujte funkci, která provádí „Shaker sort“ představený na přednášce. Pro úplnost přidávám pseudokód (Algoritmus 1) převzatý právě z přednášky a připomínám, že použité značení pro **for** cyklus počítá s průchodem do horní meze včetně.

Pro **Swap** použijte buď svoji vlastní funkci nebo `std::swap`. Pro vstupní pole k setřizení je doporučeno použít `std::vector`.

Až budete mít hotovou implementaci, porovnejte výsledek své funkce s tím, co dostanete z `std::swap`. Pomocí následující funkce vytvořte náhodný vektor a vypusťte na něj svou funkci a `std::swap`.

```
1 #include <vector>
2 #include <random>
3
4 std::vector<int> generateRandomVector(int length, int minVal, int maxVal) {
5     std::random_device rd;
6     std::mt19937 gen(rd());
7     std::uniform_int_distribution<int> distribution(minVal, maxVal);
8
9     std::vector<int> result;
10    result.reserve(length); // better then series of push_backs
11
12    for (int i = 0; i < length; ++i) {
13        result.push_back(distribution(gen));
14    }
15    return result;
16 }
```

¹Připomínám, že to znamená najít explicitní předpis výpočtu $T(n)$ pomocí n a napsat jako matematickou funkci, např. $T(n) = 2n + 5$.

Pomocí následujícího pak změřte čas pro jednotlivé sorty v závislosti na délce vektoru. Takhle to vypadá pro knihovní sort, pro Váš to bude obdobné.

```

1 #include <algorithm>
2 #include <chrono>
3 #include <iostream>
4
5 auto start = std::chrono::high_resolution_clock::now();
6 std::sort(randomVector.begin(), randomVector.end());
7 auto end = std::chrono::high_resolution_clock::now();
8 std::chrono::duration<double> duration = end - start;
9 std::cout << "Time taken to sort the vector: " << duration.count() << "
10   seconds" << std::endl;

```

A Pseudokód

Algorithm 1 shakerSort(A)

```

1:  $Left \leftarrow 0$ ;
2:  $Right \leftarrow n - 1$ ;
3: repeat
4:   LEFTTORIGHT( $A, Left, Right$ );
5:   RIGHTTOLEFT( $A, Left, Right$ );
6: until  $Left \geq Right$ ;

```

Algorithm 2 leftToRight($A, Left, Right$)

```

1:  $j \leftarrow 0$ ;
2: for  $i \leftarrow Left$  to  $Right - 1$  do
3:   if  $A[i] > A[i + 1]$  then
4:     Swap( $A[i], A[i + 1]$ );
5:      $j \leftarrow i$ ;
6:   end if
7: end for
8:  $Right \leftarrow j$ ;

```

Algorithm 3 rightToLeft($A, Left, Right$)

```

1:  $j \leftarrow 0$ ;
2: for  $i \leftarrow Right$  downto  $Left + 1$  do
3:   if  $A[i - 1] > A[i]$  then
4:     Swap( $A[i - 1], A[i]$ );
5:      $j \leftarrow i$ ;
6:   end if
7: end for
8:  $Left \leftarrow j$ ;

```
