

Структури от данни и програмиране

1: Сложност

доц. Атанас Семерджиев

доц. Петър Армянов

1

Съдържание

- Времева и пространствена сложност
- Видове мярка
- Locality
- Анализ на поведението на програмата

2

2

Времева и пространствена сложност

- Какво означава „времева“ сложност и може ли времето да е мярка за сложност.
- Какво означава „пространствена сложност“.
- Защо емпиричното измерване никога не е лесно.
- Profiler
- Locality: <https://www.youtube.com/watch?v=kcdaiPb7vaY>

3

3

Сложности (нотация на Bachmann–Landau)

Сложност	Дефиниция
$f(x) = O(g(x))$	$\exists(N > 0)\exists x_0\forall(x > x_0)(f(x) \leq Ng(x))$
$f(x) = o(g(x))$	$\forall(N > 0)\exists x_0\forall(x > x_0)(f(x) \leq N g(x))$
$f(x) = \Theta(g(x))$	$\exists(N > 0)\exists(M > 0)\exists x_0\forall(x > x_0)(Ng(x) \leq f(x) \leq Mg(x))$
$f(x) = \Omega(g(x))$	$\exists(N > 0)\exists x_0\forall(x > x_0)(f(x) \geq Ng(x))$

* Възможно е да срещнете и други дефиниции за нотациите

4

4

Някои бележки

- Complexity class vs Big-O notation
- Курсът обхваща класическите алгоритми и структури от данни и техните сложности. В него НЕ СЕ включват метрики като например:
 - Cyclomatic complexity
 - Branch complexity

5

5

Случаи

- Best-case
- Average-case
- Amortized
- Worst-case

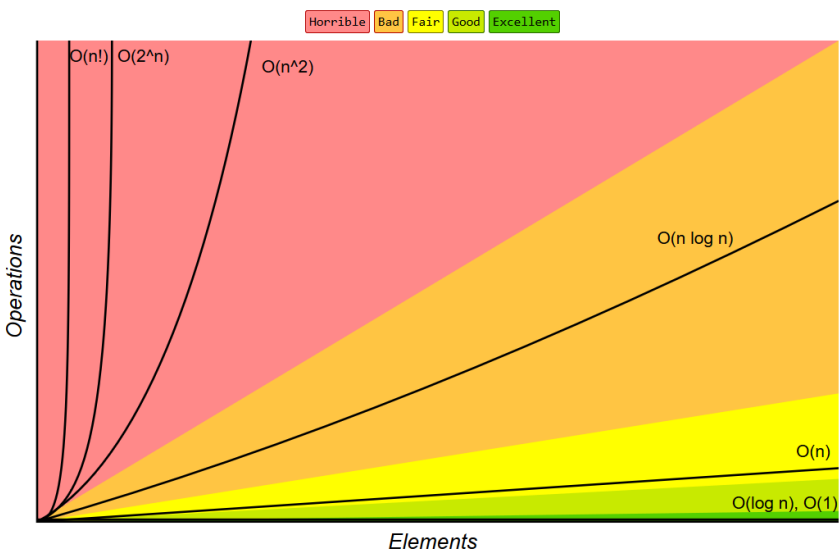
За повече информация вижте например: https://en.wikipedia.org/wiki/Best,_worst_and_average_case

6

6

Big-O

- $O(N!)$
- $O(2^N)$
- $O(N^2)$
- $O(N \log N)$
- $O(N)$
- $O(\log N)$
- $O(1)$



* Източник на графиката: Big-O Cheat Sheet - Know Thy Complexities! <http://bigocheatsheet.com/>

Big-O in practice

	1	2	3	10	100	1000	1 000 000
$O(1)$	1	1	1	1	1	1	1
$O(\log N)$	1	1	2	4	7	10	20
$O(N)$	1	2	3	10	100	1000	1 000 000
$O(N \log N)$	1	2	5	34	665	9964	19 931 569
$O(N^2)$	1	4	9	100	10 000	1 000 000	10^{12}
$O(2^N)$	1	4	8	1024	$> 10^{30}$	$> 10^{301}$	$> 10^{301029}$
$O(N!)$	1	2	6	3 628 800	$> 10^{157}$	$> 10^{2567}$	$> 10^{5565708}$

Бележки

- Това, че два алгоритъма са в един клас не значи, че единият няма да е по-добър от другия в даден случай.
- Това, че един алгоритъм е в „по-добър“ клас от друг, не значи, че той ще е по-добър във всички случаи.
- Преди да можем да дадем оценка за сложността, трябва да знаем за кой изчислителен модел става въпрос.

9

9

Тестване

- Unit Testing
 - On individual pieces of code
- Integration Testing
- Acceptance Testing
 - The requirements of the specification/contract are met
- User Acceptance Testing (UAT)
 - The user accepts the solution (utility)
- Build Verification Testing (BVT) / Build Acceptance Testing (BAT)
 - Verify that the build is testable
 - (e.g. all components have been built, data files are included, components are properly integrated, etc.)

10

10