Домашна работа

Сложност на алгоритъм

•За да можем да дефинираме понятието злобност на алгоритъм ще трябва да направим важни стъпки:

•Да фиксирамемножеството от възможни стойности на параметрите на М3.

•**Да фиксираме изчислителния формализъм - операции, начин на предаване на процедурите (синтаксис и семантика).**

•Да определим изключително важното за теорията понятие размер на входа.

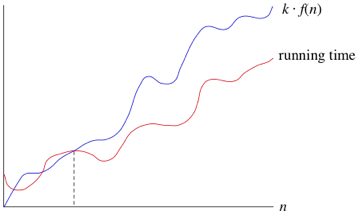
**Нотация Big-O**

**Използваме нотация big-Θ за да ограничим асимптотично отгоре и отдолу растежа на времето за изпълнение в рамките на константните коефициенти. Понякога искаме да ограничаваме само отгоре.**

**Например въпреки че времето за изпълнение на двоично търсене в най-лошия случай е Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis, би било неправилно, ако заявим, че двоичното търсене се изпълнява за време Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis във всички случаи. Ами ако открием търсената стойност с първото си предположение? Тогава алгоритъмът се изпълнява за време Θ(1)Θ(1)\Theta, left parenthesis, 1, right parenthesis. Времето за изпълнение на двоичното търсене никога не е по-бавно от  Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis, но понякога е по-бързо.**

**Би било удобно, ако имаме вид асимптотична нотация, която означава, че "времето за изпълнение нараства най-много толкова, но може бавно да порасне още". Използваме "big-O" нотация за точно такива случаи.**

**Ако времето за изпълнение е O(f(n))O(f(n))O, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis, то за достатъчно голямо nnn времето за изпълнение е най-много k⋅f(n)k⋅f(n)k, dot, f, left parenthesis, n, right parenthesis за константа kkk. Ето как да мислим за време O(f(n))O(f(n))O, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis:**

****

**6n^2 vs 100n+300**

**Казваме, че времето за изпълнение е "голямо O за f(n)f(n)f, left parenthesis, n, right parenthesis" или само "O за f(n)f(n)f, left parenthesis, n, right parenthesis." Използваме нотацията big-O за асимптотични горни граници, тъй като това дава горна граница на нарастването на времето за изпълнение за достатъчно големи входни данни.**

**Сега имаме начин да характеризираме времето за изпълнение на двоичното търсене при всички случаи. Можем да кажем, че времето за изпълнение е винаги O(2n)O(log2​n)O, left parenthesis, log, start base, 2, end base, n, right parenthesis. Можем дори по-уверено да кажем какво ще е времето в най-лошия случай: ще е Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis. Но ако искаме да обхванем всички случаи, можем да кажем, че двоичното търсене се изпълнява за време O(2n)O(log2​n)O, left parenthesis, log, start base, 2, end base, n, right parenthesis.**

**Ако се върнеш към определението за нотация big-Θ, ще забележиш, че изглежда много подобно на това за нотацията big-O, само че при big-Θ времето има не само горна, но и долна граница. Ако кажем, че времето за изпълнение е Θ(f(n))Θ(f(n))\Theta, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis в дадена ситуация, то е също и O(f(n))O(f(n))O, left parenthesis, f, left parenthesis, n, right parenthesis, right parenthesis. Например, можем да заключим, че тъй като времето за изпълнение на двоичното търсене в най-лошия случай е Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis, то е също и O(2n)O(log2​n)O, left parenthesis, log, start base, 2, end base, n, right parenthesis.**

**Обратното не е задължително вярно: както видяхме, можем да кажем, че двоичното търсене винаги се изпълнява за време O(2n)O(log2​n)O, left parenthesis, log, start base, 2, end base, n, right parenthesis, но не винаги се изпълнява за време Θ(2n)Θ(log2​n)\Theta, left parenthesis, log, start base, 2, end base, n, right parenthesis.**

**Тъй като нотацията big-O дава само асимптотична горна граница, а не асимптотично твърда граница, можем да изказваме твърдения, които на пръв поглед изглеждат неверни, но са технически верни. Например напълно прави бихме били, ако кажем, че двоичното търсене се изпълнява за време O(n)O(n)O, left parenthesis, n, right parenthesis. Това е така, защото времето за изпълнение не нараства по-бързо от константно количество nnn пъти. Всъщност, расте по-бавно.**

**Помисли за това така. Представи си, че имаш 10 долара в джоба си. Отиваш при другарче и казваш: "Имам някакво количество пари в джоба си и гарантирам, че е не повече от един милион долара." Твоето твърдение е напълно вярно, но не особено прецизно.**

**Един милион долара е горна граница за 10 долара, също както O(n)O(n)O, left parenthesis, n, right parenthesis е горна граница за времето за изпълнение на двоичното търсене. Други не така прецизни горни граници биха били O(n2)O(n2)O, left parenthesis, n, squared, right parenthesis, O(n3)O(n3)O, left parenthesis, n, cubed, right parenthesis и O(2n)O(2n)O, left parenthesis, 2, start superscript, n, end superscript, right parenthesis. Но нито една от Θ(n)Θ(n)\Theta, left parenthesis, n, right parenthesis, Θ(n2)Θ(n2)\Theta, left parenthesis, n, squared, right parenthesis, Θ(n3)Θ(n3)\Theta, left parenthesis, n, cubed, right parenthesis и Θ(2n)Θ(2n)\Theta, left parenthesis, 2, start superscript, n, end superscript, right parenthesis не биха описали правилно времето за изпълнение на двоичното търсене във всеки случай.**