

СЛОЖНОСТ НА АЛГОРИТМИЧНИ ЗАДАЧИ

КОНТРОЛНО № 5 ПО “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ” — СУ, ФМИ, 2019 Г.

Задача 1. Формулираме алгоритмичната задача `PointsInPlane` така: n точки в равнината, зададени с координатите си, да се наредят по нарастващ ред на разстоянието си до т. $O(0; 0)$.

Докажете, че ако се решава чрез сравнения, задачата има времева сложност $\Theta(n \log n)$. За целта:

а) съставете алгоритъм със сложност $O(n \log n)$; (5 точки)

б) докажете долната граница $\Omega(n \log n)$ чрез подходяща редукция. (15 точки)

Задача 2. Нека $f(n)$ е времевата сложност на задачата за умножение на две n -цифрени числа, а $g(n)$ е времевата сложност на задачата за намиране на квадрата на n -цифрено число. Тези две сложности са неизвестни поотделно. Докажете, че $g(n) = \Theta(f(n))$. За целта:

а) докажете, че $f(n) = \Omega(n)$ и $g(n) = \Omega(n)$; (5 точки)

б) докажете, че $g(n) = O(f(n))$; (5 точки)

в) докажете, че $g(n) = \Omega(f(n))$. (10 точки)

Задача 3. Разглеждаме следната задача за разпознаване, да я наречем `Problem3`:

Вход: неориентиран нетегловен граф G с n върха и m ребра.

Въпрос: Може ли върховете на графа да се номерират с целите числа от 1 до n така, че номерата на двата края на всяко ребро да не са последователни цели числа?

Докажете, че задачата `Problem3` е NP-трудна. (20 точки)

РЕШЕНИЯ

Задача 1. а) Алгоритъм със сложност $O(n \log n)$: сортираме точките по разстояние до т. $O(0; 0)$ с помощта на някоя от бързите сортировки, например чрез пирамидално сортиране. Разстоянията се пресмятат по известната формула от геометрията. Пресмятането на всяко отделно разстояние става за константно време; на всички разстояния общо — за време $\Theta(n)$. Същинското сортиране се извършва за време $\Theta(n \log n)$. Окончателно, времето на целия алгоритъм е $\Theta(n \log n)$. Ето защо задачата `PointsInPlane` може да се реши за време $O(n \log n)$.

б) Долната граница $\Omega(n \log n)$ на задачата `PointsInPlane` се доказва с помощта на редукция от задачата `SortPositive` за сортиране на масив $A[1..n]$ от положителни числа, за която знаем, че изисква време $\Omega(n \log n)$, когато се решава чрез сравнения.

```
SortPositive(A[1...n])
P[1...n]: array of points in the plane
for k ← 1 to n do
    P[k].x ← A[k]
    P[k].y ← 0
PointsInPlane(P[1...n])
for k ← 1 to n do
    A[k] ← P[k].x
```

Коректност на редукцията: Точката $P[k]$ изобразява числото $A[k]$ върху положителната част на абсцисната ос, затова разстоянието от т. $P[k]$ до т. O е равно на $A[k]$. Функцията `PointsInPlane` подрежда точките по разстоянията им от нулата, тоест подрежда числата $A[k]$ в нарастващ ред.

Бързина на редукцията: Двата цикъла имат обща времева сложност $\Theta(n) = o(n \log n)$, затова редукцията е достатъчно бърза за целите на доказателството.

Задача 2. а) Това са тривиални долни граници по размера на входа: всеки алгоритъм трябва поне да прочете цифрите на числата.

б) Повдигането на квадрат е частен случай на умножение. Частният случай е не по-труден от общия, затова $g(n) = O(f(n))$.

в) Обратно, $g(n) = \Omega(f(n))$, тоест $f(n) = O(g(n))$, защото умножението може да се сведе до повдигане на квадрат по формулата $xy = \frac{(x+y)^2 - (x-y)^2}{4}$. Останалите аритметични операции

в тази формула — делението на 4, събирането и изваждането — изискват време $\Theta(n)$. Ето защо тази формула представлява алгоритъм за умножение на две n -цифрени числа с времева сложност $\Theta(g(n)) + \Theta(n) = \Theta(g(n))$, защото $g(n) = \Omega(n)$ според доказаното в подусловие “а”. А щом има алгоритъм за умножение със сложност $\Theta(g(n))$, тогава времевата сложност на умножението е $f(n) = O(g(n))$.

Задача 3. Извършваме редукция от известната NP-пълна задача Хамилтонов_път:

Вход: неориентиран нетегловен граф G с n върха и m ребра.

Въпрос: Графът G съдържа ли хамилтонов път?

Описание на редукцията:

Хамилтонов_път (G : неориентиран нетегловен граф)

- 1) Образоваме допълнението на множеството от ребрата на графа G , тоест изтриваме наличните ребра и добавяме липсващите.
- 2) **return** Problem3 (G)

Коректност на редукцията:

Функцията Хамилтонов_път (от първоначалния граф G) връща стойност “истина”.

⇕ (от ред № 2 на алгоритъма)

Функцията Problem3 (от променения граф G) връща стойност “истина”.

⇕ (от определението на задачата Problem3)

Върховете на променения граф G могат да се номерират с целите числа от 1 до n така, че всеки два върха, номерирани с последователни цели числа, да не са свързани с ребро.

⇕ (от ред № 1 на алгоритъма)

Върховете на първоначалния граф G могат да се номерират с целите числа от 1 до n така, че всеки два върха, номерирани с последователни цели числа, да са свързани с ребро.

⇕ (от определението за хамилтонов път)

Първоначалният граф G съдържа хамилтонов път.

С това е доказано, че функцията Хамилтонов_път (G) връща стойност “истина” \Leftrightarrow неориентираният нетегловен граф G съдържа хамилтонов път. Това точно съответства на постановката на задачата Хамилтонов_път. Следователно редукцията е коректна.

Бързина на редукцията: Редукцията се състои само от образуването на допълнението на множеството от ребрата на графа. Така за всеки връх се обработват $n - 1$ ребра, а общо за целия граф броят на операциите добавяне и изтриване на ребро е равен на $n(n - 1) = n^2 - n = \Theta(n^2)$. Тоест времевата сложност на редукцията е квадратична, следователно полиномиална.

ТОЧКУВАНЕ

Задача 1.

- а) 5 точки за описание на алгоритъм с времева сложност $O(n \log n)$;
- б) 15 точки за доказване на долната граница, от които по 5 точки за всяка от следните три стъпки:
 - описание на редукцията;
 - доказателство за коректност на редукцията;
 - анализ на бързината на редукцията.

Задача 2.

- а) 5 точки за позоваване на тривиалните долни граници по размера на входа;
- б) 5 точки за позоваване на връзката между общия и частния случай;
- в) 10 точки, от които по 5 точки за всяка от следните две стъпки:
 - свеждане на умножението до повдигане на квадрат;
 - анализ на бързината на алгоритъма.

Задача 3 носи 20 точки, в това число:

- описание на редукцията: 7 точки;
- доказателство за коректност на редукцията: 7 точки;
- анализ на бързината на редукцията: 6 точки.