И З С Л Е Д В А Н Е   Н А   А Л Г О Р И Т М И

ПРИМЕРНО КОНТРОЛНО № 1 ПО “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ” — СУ, ФМИ

(ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”, 1. ПОТОК, ФЕВРУАРИ – ЮНИ 2019 Г.)

Задача 1. Нека $f(n)$ и $g(n)$ са асимптотично положителни функции, за които $f(n) \succ g(n)$. Докажете, че $f(n) - g(n) \asymp f(n)$. Покажете с контрапример, че ако неравенството е нестрого, твърдението ще престане да бъде вярно.

Задача 2. Намерете порядъка на обратната функция на факториела:

$w(n) =$ най-голямото цяло положително число k, за което $k! \leq n$.

Функцията $w(n)$ е определена за цели положителни стойности на n. Например

$$w(1) = 1,\ \ w(2) = w(3) = w(4) = w(5) = 2,\ \ w(6) = \cdots = w(23) = 3,\ \ w(24) = 4.$$

Задача 3. Да се реши рекурентното уравнение $T(n) = T\left(\dfrac{n}{5}\right) + T\left(\dfrac{n}{8}\right) + n^3.$

Задача 4. Разглеждаме два алгоритъма за разпознаване на прости числа — `isPrime1` и `isPrime2`. Входът им е цяло положително число `n ≥ 2`.

```
isPrime1(n)  // n ≥ 2
for k ← 2 to n-1 do
  if n се дели на k
    return false
return true
```

```
isPrime2(n)  // n ≥ 2
return not hasDivisor(n, 2, n-1)

hasDivisor(n, k, r)
// проверява дали n има делител
// между k и r включително (k ≤ r)
if k = r
  if n се дели на k
    return true
  else
    return false
if k + 7 > r
  for j = k to r do
    if n се дели на j
      return true
  return false
for j = 1 to 6 do
  if hasDivisor(n, k+(j-1)⌊(r-k)/7⌋, k+j⌊(r-k)/7⌋)
    return true
return hasDivisor(n, k+6⌊(r-k)/7⌋, r)
```

а) Докажете с инварианта, че `isPrime1` е коректен алгоритъм.

б) Докажете по индукция, че hasDivisor е коректен алгоритъм.

в) Намерете сложността по време на алгоритъма `isPrime2`.

РЕШЕНИЯ

Задача 1. $f(n) \succ g(n) \Rightarrow \lim\limits_{n\to\infty} \dfrac{g(n)}{f(n)} = 0$

$$\Rightarrow \lim_{n\to\infty} \frac{f(n)-g(n)}{f(n)} = \lim_{n\to\infty}\left(1-\frac{g(n)}{f(n)}\right) = 1-0 = 1 \Rightarrow f(n)-g(n) \asymp f(n).$$

15 точки

Ако асимптотичното неравенство е нестрого, то ще допуска възможността за асимптотично равенство на двете функции, тоест $g(n) \asymp f(n)$. Да вземем $f(n)=n^2+7n$ и $g(n)=n^2+2n$. Тогава $f(n)-g(n) = 5n \prec f(n)=n^2+7n$, следователно не е вярно, че $f(n)-g(n) \asymp f(n)$. **5 точки**

Задача 2. От определението на функцията w следва, че $w(n) \to \infty$ при $n \to \infty$ и че е изпълнено следното двойно неравенство:

$$w! \leq n < (w+1)!.$$

Логаритмуваме:

$$w \log w \asymp \log\big(w!\big) \leq \log n < \log\big((w+1)!\big) \asymp (w+1)\log(w+1) \asymp w \log w.$$

От неравенствата $w \log w \preceq \log n \preceq w \log w$ следва, че $\log n \asymp w \log w$.

Затова $w \asymp \dfrac{\log n}{\log w}$. **4 точки**

Развиваме това уравнение, но само с една стъпка:

$$w \asymp \frac{\log n}{\log\log n - \log\log w}.$$

4 точки

(Ако извършим повече стъпки, знаменателят ще се усложни.)

Прилагаме свойството от задача 1: тъй като $\log\log n \succ \log\log w$, то знаменателят се опростява до $\log\log n$, тоест

$$w(n) \asymp \frac{\log n}{\log\log n}.$$

4 точки

Това е отговорът на задачата — порядъкът на обратната функция на факториела.

Обаче в горните разсъждения използвахме без доказателство неравенството

$$\log\log n \succ \log\log w.$$

Доказателство: По-горе установихме, че $\log n \asymp w\log w$, следователно $\log\log n \asymp \log(w\log w) = \log w + \log\log w \asymp \log w \succ \log\log w$. **4 точки**

Остава да докажем неравенството

$$\log w \succ \log\log w,$$

което се използва два пъти в горната редица от асимптотични сравнения:

1) самостоятелно — в последната стъпка;
2) като основание за пренебрегване на събираемото от по-нисък порядък — в предпоследната стъпка.

Неравенството $\log w \succ \log\log w$ се доказва чрез граничен преход, в който се използва правилото на Лопитал:

$$\lim_{w\to\infty}\frac{\ln w}{\ln\ln w} = \left[\frac{\infty}{\infty}\right] = \lim_{w\to\infty}\frac{(\ln w)'}{(\ln\ln w)'} =$$

$$= \lim_{w\to\infty}\frac{\dfrac{1}{w}}{\dfrac{1}{\ln w}\cdot\dfrac{1}{w}} = \lim_{w\to\infty}\frac{1}{\dfrac{1}{\ln w}} = \lim_{w\to\infty}\ln w = \infty.$$ **4 точки**

Задача 3 може да се реши по индукция или чрез развиване на уравнението и изследване на дървото на рекурсията. Получава се $T(n) \asymp n^3$. **20 точки**

Задача 4.

а) Инварианта на цикъла на `isPrime1`:

`n` не се дели на никое от числата 2, 3, ... , `k` – 1. **15 точки**

Точкуване: 5 т. за формулиране на инвариантата; 5 т. за нейното доказване; 5 т. за доказване на коректност на алгоритъма.

б) Доказва се със силна индукция по `r` – `k`. **15 точки**

Точкуване: 5 т. за базата на индукцията; 5 т. за индуктивната стъпка; 5 т. за доказване, че алгоритъмът ще завърши.

в) Времевата сложност $T(n)$ на алгоритъма `isPrime2`

удовлетворява уравнението $T(n) = 7T\left(\dfrac{n}{7}\right) + 1$, **5 точки**

чието решение се получава от мастър-теоремата: $T(n) = \Theta(n)$. **5 точки**