АНАЛИЗ НА АЛГОРИТМИ

КОНТРОЛНО № 1 ПО "ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ" — СУ, ФМИ, 2019 Г.

Задача 1. Намерете стойността, върната от алгоритъма `F1`. Дайте строга и пълна обосновка с помощта на инварианта.

```
F1(A[1...n])
b ← 0
for k ← 1 to n
  if A[k] = 4
    b ← b + 1
return b
```

(5 точки)

Задача 2. Намерете с обосновка порядъка на времевата сложност на алгоритъма `F2`.

(5 точки)

```
F2(A[1...n])
if n < 3
  return F1(A[1...n])
b ← 0
for k ← 1 to 2
  p ← F2(A[k...n+k-2])
  q ← F2(A[k...n+k-3])
  b ← b + p + q
b ← b + F2(A[3...n])
return b
```

Задача 3. Намерете с обосновка порядъка на времевата сложност на алгоритъма `F3`.

(5 точки)

```
F3(A[1...n])
if n < 16
  return F1(A[1...n])
r ← ⌊n/2⌋ - 1
b ← F3(A[n-r...n])
for k ← 1 to 7
  p ← F3(A[k...r+k])
  b ← b + p
return b
```

Задача 4. Подредете по асимптотичен порядък следните функции: **(5 точки)**

$$n^2 10^{n^7}, \quad 7^{n^{10}} \log^5 n, \quad \log^8 n, \quad (3n)!, \quad 78n^9.$$

ТОЧКУВАНЕ

Задача 1. Всяка от следните пет стъпки носи по 1 точка:
— формулиране на вярна и използваема инварианта;
— база на доказателството на инвариантата;
— индуктивна стъпка на доказателството на инвариантата;
— доказателство, че алгоритъмът `F1` ще завърши (брой изпълнения на тялото на цикъла);
— получаване на върнатата стойност като следствие от инвариантата.

Задача 2 носи 2 т. за съставяне на линейно-рекурентно уравнение и 3 т. за решаването му, като тези 3 точки са разпределени по следния начин: 1 точка за намиране на корените на съответното характеристично уравнение; 1 точка за намиране на общото решение на рекурентното уравнение; 1 точка за намиране на асимптотичния порядък на решението, опростен възможно най-много.

Задача 3 носи 2 точки за съставяне на рекурентно уравнение и 3 точки за решаването му с помощта на мастър-теоремата (или по друг подходящ начин).

Задача 4 носи 1 точка за подреждане на петте функции с помощта на асимптотичния ред и още 4 точки за доказателство на полученото подреждане — по 1 точка за всяко доказано сравнение на съседни функции. Дават се точки само за сравнения на съседни функции. Ако в подреждането на функциите (тоест в отговора) има грешка, не се дават никакви точки.

Забележка: Не се дават точки за решаването на грешно съставено рекурентно уравнение.

Р Е Ш Е Н И Я

Задача 1. Инвариантата на цикъла на алгоритъма `F1` гласи:

Всеки път, когато се изпълнява проверката за край на цикъла `k` $\overset{?}{\leq}$ `n`,
стойността на променливата `b` е равна на броя на четворките в подмасива `A[1...k-1]`.

Инвариантата се доказва с помощта на математическа индукция.

База: Проверката за край на цикъла `k` $\overset{?}{\leq}$ `n` се изпълнява за първи път, когато алгоритъмът влиза в цикъла. В този момент променливата `k` има стойност 1 и подмасивът `A[1...k-1]` има `k-1` = 0 елемента, тоест подмасивът е празен. Следователно той не съдържа четворки, тоест броят на четворките в подмасива `A[1...k-1]` е равен на 0, колкото е началната стойност на променливата `b`. Затова при първата проверка за край на цикъла е изпълнено равенството, твърдяно от инвариантата.

Индуктивна стъпка: Нека при някоя проверка за край на цикъла, която не е последна, стойността на променливата `b` е равна на броя на четворките в подмасива `A[1...k-1]`. Ще докажем, че това равенство остава в сила и при следващата проверка за край на цикъла. Действително, щом текущата проверка не е последна, това означава, че тялото на цикъла се изпълнява поне още веднъж. Според индуктивното предположение стойността на `b` в началото на тялото на цикъла е равна на броя на четворките в подмасива `A[1...k-1]`. Ако `A[k] = 4`, стойността на `b` се увеличава с 1, но и броят на четворките нараства с 1. В противен случай стойността на `b` остава същата, но и броят на четворките не се променя. И в двата случая стойността на `b` е равна на броя на четворките в подмасива `A[1...k]` след изпълнение на тялото на цикъла. После стойността на `k` се увеличава с 1, поради което същият подмасив вече се записва като `A[1...k-1]`, тоест стойността на променливата `b` е равна на броя на четворките в подмасива `A[1...k-1]` (поради новата стойност на `k` това е новият подмасив, въпреки че се записва по стария начин). След това алгоритъмът преминава към проверката за край на цикъла. Току-що доказахме, че в този миг стойността на променливата `b` е равна на броя на четворките в подмасива `A[1...k-1]`. Тъкмо това гласи инвариантата, така че тя е в сила и при новата проверка за край на цикъла.

С това инвариантата е доказана.

Завършек: Понеже цикълът е по брояч от 1 до `n`, то алгоритъмът ще завърши след точно `n` изпълнения на тялото на цикъла. При последната проверка за край на цикъла неговият брояч има стойност `k = n+1`. Инвариантата важи и при последната проверка, затова имаме право да заместим `k = n+1` в инвариантата. Получаваме, че при излизане от цикъла стойността на `b` е равна на броя на четворките в подмасива `A[1...k-1]`, тоест `A[1...n]`, което всъщност е целият входен масив. Именно тази стойност се връща от последната команда на алгоритъма.

Отговор: Алгоритъмът `F1` връща броя на четворките в масива `A[1...n]`.

Задача 2. Изпълняват се две рекурсивни извиквания върху подмасиви с дължина `n-1` и три рекурсивни извиквания върху подмасиви с дължина `n-2`. Всички останали действия изразходват общо константно време, тъй като тялото на цикъла се изпълнява само два пъти. Затова времевата сложност $\mathrm{T(n)}$ на алгоритъма `F2` удовлетворява рекурентното уравнение

$$\mathrm{T(n)} = 2\,\mathrm{T(n-1)} + 3\,\mathrm{T(n-2)} + \Theta(1).$$

Това е линейно-рекурентно уравнение. Съответното му характеристично уравнение е

$$\lambda^{n} = 2\lambda^{n-1} + 3\lambda^{n-2}$$

Ненулевите корени на характеристичното уравнение намираме, като го разделим на λ^{n-2}:

$$\lambda^{2} = 2\lambda + 3 \iff \lambda^{2} - 2\lambda - 3 = 0.$$

Корените на това квадратно уравнение са $\lambda_1 = 3$ и $\lambda_2 = -1$. От свободния член $1 = \mathrm{n}^0\, 1^{\mathrm{n}}$ на рекурентното уравнение идва още един характеристичен корен — числото 1 (основата на показателната функция). Мултимножеството от характеристични корени е $\{\, 3;\ 1;\ -1 \,\}_{\mathrm{m}}$ и общото решение на рекурентното уравнение е

$$\mathrm{T(n)} = C_1\, 3^{\mathrm{n}} + C_2\, 1^{\mathrm{n}} + C_3\, (-1)^{\mathrm{n}}.$$

Второто и третото събираемо са ограничени функции $\Theta(1)$, затова само първото събираемо е определящо за порядъка на времевата сложност. Следователно $\mathrm{T(n)} = \Theta\left(3^{\mathrm{n}}\right)$.

Забележка: Свободният член на линейно-рекурентните уравнения е важен поначало, но тук е ограничена функция $\Theta(1)$ и може да се пренебрегне: ако запишем уравнението във вида: $\mathrm{T(n)} = 2\,\mathrm{T(n-1)} + 3\,\mathrm{T(n-2)}$, това няма да промени асимптотиката на решението.

Задача 3. Изпълняват се осем рекурсивни извиквания върху подмасиви с дължина

$$\mathtt{r + 1} = \lfloor \mathtt{n / 2} \rfloor.$$

Другите действия изразходват константно време, защото тялото на цикъла се изпълнява само седем пъти. Времевата сложност $\mathrm{T(n)}$ на алгоритъма `F3` удовлетворява уравнението

$$\mathrm{T(n)} = 8\,\mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + \Theta(1).$$

От първия случай на мастър-теоремата намираме порядъка на решението: $\mathrm{T(n)} = \Theta\left(\mathrm{n}^3\right)$.

Задача 4. Дадените функции се подреждат по асимптотичен порядък така:

$$\log^8 n \prec 78\,n^9 \prec (3\,n)! \prec n^2\, 10^{n^7} \prec 7^{n^{10}} \log^5 n.$$

Първото сравнение се доказва така: коренуваме го с показател 8, след което използваме граничен преход и теоремата на Лопитал. Останалите три сравнения след логаритмуване се свеждат до редицата от сравнения

$$\log n \prec n \log n \prec n^7 \prec n^{10},$$

които вече се доказват чрез граници на отношенията на всеки две съседни функции.