Analiza Algorytmów, Lista 6 Raport

Tomasz Krent

18 czerwca 2020

1 Zadanie 14

Patrząc na algorytm z zadania 14 pod względem liczby wywołań linii 6 w danym algorytmie układamy i rozważamy następujące zadanie rekurencyjne: $a_0 = 0$,$a_n = n + \sum_{i=0}^{n-1} a_i$, gdzie a_i oznacza ilość wywołań linii 6 dla danej wejściowej i. Rozważając rekurencję i wykorzystując funkcję tworzącą otrzymujemy:

$a_{n-1} = n - 1 + \sum_{i=0}^{n-2} a_i$
$a_n = 1 + a_{n-1} + n - 1 + \sum_{i=0}^{n-2} a_i = 1 + 2_{n-1}$
$A(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 x^0 + \sum_{n=1}^{\infty} a_n x^n = a_0 x^0 + \sum_{n=1}^{\infty} 2a_{n-1} x^n + \sum_{n=1}^{\infty} 1x^n = 2x \sum_{n=1}^{\infty} a_{n-1} x^{n-1} + \sum_{n=1}^{\infty} x^n = 2A(x) + \sum_{n=1}^{\infty} x^n$
$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x} \Rightarrow 1 + \sum_{n=1}^{\infty} x^n = \frac{1}{1-x} \Rightarrow \sum_{n=1}^{\infty} x^n = \frac{x}{1-x}$
$A(x) = A(x) + \frac{x}{1-x} \Rightarrow A(x) - 2xA(x) = \frac{x}{1-x} \Rightarrow A(x) = \frac{x}{(1-2x)(1-x)} = \frac{1}{1-2x} - \frac{1}{1-x} = \sum_{n=0}^{\infty} (2x)^n - \sum_{n=0}^{\infty} x^n = \sum_{n=0}^{\infty} (2^n - 1)x^n$
$a_n = 2^n - 1$

Wyniki działania programu *zad*14.*py* sprawdzającego eksperymentalnie ilość wywołań linii numer 6 w algorytmie z zadania 14 w porównaniu z przewidzianymi wynikami teoretycznymi:

n	wyniki eskperymentalne	wyniki teoretyczne
0	0	0
1	1	1
2	3	3
3	7	7
4	15	15
5	31	31
6	63	63
7	127	127
8	255	255

Przewidywana liczba wywołań linii 6 algorytmu zgadza się z wynikami programu wyznaczonymi w sposób ekperymentalny.

2 Zadanie 15

Układamy i rozważamy następujące równanie rekurencyjne: $b_0 = 1$,$b_1 = 1$, $b_2 = 1 + k \cdot b_1 + k_2$, $b_n = 1 + \sum_{i=0}^{n} k_i \cdot b_i$, gdzie $b_i, i \in \{0..n\}$ oznacza średnią ilość wywołań algorytmu dla danego i, a k_i oznacza prawdopodobieństwo, że algorytm dla danego i zostanie wywołany, w tym przypadku $k_i = \frac{1}{2}$. Rozważając rekurencję i wykorzystując funkcję tworzącą otrzymujemy:

$b_2 = 1 + \frac{1}{2} \cdot 1 + \frac{1}{2} b_2 \Rightarrow \frac{1}{2} b_2 = \frac{3}{2} \Rightarrow b_2 = 3$
$b_{n-1} = 1 + \sum_{i=0}^{n-1} \frac{1}{2} \cdot b_i$
$b_n = 1 + \frac{1}{2} b_n \sum_{i=0}^{n-1} \frac{1}{2} \cdot b_i$
$b_n = \frac{1}{2} b_n + b_{n-1}$
$b_n = 2 b_{n-1}$
$b_3 = 2 \cdot 3$
$b_4 = 2 \cdot 2 \cdot 3$
$b_5 = 2 \cdot 2 \cdot 2 \cdot 3$
$\vdots$
$b_n = \underbrace{2 \cdot 2 \cdot .. \cdot 2}_{n-2} \cdot 3$
$b_n = 2^{n-2} \cdot 3$, dla $n \geqslant 2$

Wyniki działania programu *zad*15.*py* sprawdzającego eksperymentalnie średnią ilość wywołań algorytmu dla danego n w 100000 próbach w porównaniu z przewidzianymi wynikami teoretycznymi:

n	wyniki eskperymentalne	wyniki teoretyczne
0	1	1
1	1	1
2	2.99028	3
3	5.9921	6
4	12.00283	12
5	24.01613	24
6	48.16056	48
7	95.98284	96
8	191.96322	192
9	384.08573	384

Przewidywana liczba wywołań algorytmu jest zbliżona do wyników uzyskanych w sposób eksperymentalny.