RECORRÊNCIAS - Aula 1 (Cormen, cap. 4)

Na aula anterior, nós concluímos que a função abaixo limita superiormente o tempo de execução do algoritmo de busca binária recursivo para um intervalo de "t" elementos:

$$f(t) = a + f(\text{piso}(t/2)), \text{ se } t > 0$$
$$f(0) = b,$$

onde "a" e "b" são constantes naturais positivas. Nós desejamos obter uma expressão não-recursiva para a função acima, a qual nos permita então exprimir o tempo de execução do algoritmo em termos da notação assintótica e de funções já bastante conhecidas.

Embora nós já conheçamos a complexidade assintótica do tempo de execução de pior caso do algoritmo de busca binária, por já termos analisado a versão iterativa do algoritmo, é instrutivo ignorar temporariamente esse conhecimento, para fins do estudo da solução de recorrências. (De qualquer forma, nós veremos mais à frente como poderíamos fazer uso direto desse conhecimento.)

Uma maneira útil de entender uma função é observar os valores assumidos por ela. No caso da função "f", temos:

t	piso(t/2)	f(t)
0	0	b
1	0	a + b
2	1	2a + b
3	1	2a + b
4	2	3a + b
5	2	3a + b
6	3	3a + b
7	3	3a + b
8	4	4a + b
9	4	4a + b
...	...	...
15	7	4a + b
16	8	5a + b
17	8	5a + b
...	...	...
31	15	5a + b
32	16	6a + b
33	16	6a + b
...	...	...
63	31	6a + b
64	32	7a + b
65	32	7a + b
...	...	...

Observando a tabela acima, e particularmente o fato de que o termo que multiplica "a" em f(t) aumenta em uma unidade a cada potência de 2 em "t", nós chegamos à seguinte conjectura:

$$f(t) = (1 + \text{piso}(\lg t))*a + b, \ \forall t > 0.$$

Tal conjectura pode ser provada por indução matemática:

```
1. LEMA (AUXILIAR): ∀t >= 2 natural, piso(lg(2*piso(t/2))) = piso(lg t).
   ---------------
================================================================================
PROVA:

Seja t >= 2 natural. Nós provaremos o resultado p/ cada um dos casos possíveis:

  * CASO 1: t = 2^k, para algum k > 0 natural. Nesse caso, temos:

      piso(lg(2*piso(t/2))) = piso(lg(2*piso((2^k)/2)))
                            = piso(lg(2*piso(2^(k-1))))
                            = piso(lg(2*2^(k-1)))
                            = piso(lg(2^k))
                            = piso(lg t), CQD.

  * CASO 2: t = 2^k + j, para certos k,j > 0 naturais tais que j < 2^k. Temos:

      * CASO 2.1: j = 2i, para algum i > 0 natural. Nesse caso, temos:

          piso(lg(2*piso(t/2))) = piso(lg(2*piso((2^k + 2i)/2)))
                                = piso(lg(2*piso(2^(k-1) + i )))
                                = piso(lg(2*(2^(k-1) + i)     ))
                                = piso(lg(2^k + 2i            ))
                                = piso(lg t), CQD.

      * CASO 2.2: j = 2i + 1, para algum i >= 0 natural. Nesse caso, temos:

          piso(lg(2*piso(t/2))) = piso(lg(2*piso((2^k + 2i + 1)/2)))
                                = piso(lg(2*piso((2^k + 2i + 1)/2)))
                                = piso(lg(2*piso(2^(k-1) +i +0,5 )))
                                = piso(lg(2*(2^(k-1) + i)        ))
                                = piso(lg(2^k + 2i               ))
                                = piso(lg(2^k + 2i               ))
                                = piso(lg(2^k + 2i               ))
                                = piso(k + x), para algum 0 < x < 1
                                = k
                                = piso(lg(2^k + j))
                                = piso(lg t), CQD.

================================================================================
```

2. LEMA: se "a" e "b" são naturais positivos e f : N -> N é a função definida
pelas equações de recorrência abaixo

```
     f(t) = a + f(piso(t/2)), se t > 0
     f(0) = b,
```

então f(t) = (1 + piso(lg t))*a + b, ∀t > 0.

==

PROVA:

Por indução matemática forte:

* BASE: t = 1: Temos:

```
      f(1) = a + f(piso(t/2))
           = a + f(piso(1/2))
           = a + f(0)
           = a + b
           = b + a*(1 + 0)
           = b + a*(1 + piso(0))
           = b + a*(1 + piso(lg 1)), CQD.
```

* PASSO: t = t' + 1, para algum t' >= 1. Temos:

```
      f(t) = a + f(piso(t/2))
           = a + b + a*(1 + piso(lg(piso(t/2)))) --> pela H.I. e já que
                                                 --> piso(t/2) > 0 (pois t >= 2)
           = b + a*(1 + 1 + piso(lg(piso(t/2))))
           = b + a*(1 + piso(1 + lg(piso(t/2))))
           = b + a*(1 + piso(lg(2) + lg(piso(t/2))))
           = b + a*(1 + piso(lg(2*piso(t/2))))
           = b + a*(1 + piso(lg t)), CQD.        --> LEMA 1
```

==

3. EXERCÍCIO: usando o lema acima, prove que f(t) = θ(lg t).

Pelo resultado acima, portanto, nós concluímos que a versão recursiva do algoritmo de busca binária executa em tempo O(lg t) (pois o tempo de execução do algoritmo é limitado superiormente por f(t), e f(t) = O(lg t)).

4. EXERCÍCIO ESSENCIAL: sejam a,b,c naturais positivos e f : N -> N a função tal que

```
     f(n) = a + f(n-1), se n > 1;
     f(1) = b;
     f(0) = c.
```

Encontre uma expressão não-recorrente para f(n), e prove a igualdade entre essa expressão e f(n) por indução matemática.