الگوریتم ساخت تمام درخت های BTS
و درستی فرمول محاسبه تعداد آنها

آرمان ریاحی

۲۰ آذر ۱۴۰۳

Algorithm 1 AllSkeleton(start, end): for sequential numbers

AllSkeleton($start$, end)
  if ($start > end$) then
    return $[null]$
  $allTrees \leftarrow [\ \ ]$
  for i **from** $start$ **to** end **do**
    $leftTrees \leftarrow$ **AllSkeleton**($start$, $i-1$)
    $rightTrees \leftarrow$ **AllSkeleton**(i + 1, end)
    for each $left$ **in** $leftTree$ **do**
      for each $right$ **in** $rightTrees$ **do**
        $currentTree \leftarrow$ **new Node**(i)
        $currentTree.left \leftarrow left$
        $currentTree.right \leftarrow right$
        add $currentTree$ **to** $allTrees$
  return $allTree$

کد پایتون مربوط به الگوریتم بالا به شکل زیر است:

```
class Node:
  def __init__(self, data):
    self.left = None
    self.right = None
    self.data = data

def allSkelet(start, end):
  if start > end :
    return [None]

  allTrees = []
  for i in range(start, end + 1):
    leftTrees = allSkelet(start, i - 1)
    rightTrees = allSkelet(i + 1, end)

    for left in leftTrees:
```

```
        for right in rightTrees:
          currentNode = Node(i)
          currentNode.left = left
          currentNode.right = right
          allTrees.append(currentNode)

    return allTrees
```

کد و الگوریتم داده شده در حالتی است که توالی لیست اعداد ما قدر نسبتی برابر با ۱ داشته باشد. الگوریتم زیر برای حالت کلی است یعنی اگر اعداد فقط حالت صعودی داشته باشند.

Algorithm 2 AllSkeleton(A[a_1, ..., a_n]): for ascending numbers

> $start = 0$
> $end = A.length - 1$
> **AllSkeleton**($start$, end)
>   **if** ($start > end$) then
>   **return** [$null$]
> $allTrees \leftarrow [\ \ ]$
>   **for** i **from** $start$ **to** end **do**
> $leftTrees \leftarrow$ **AllSkeleton**($start$, $i - 1$)
> $rightTrees \leftarrow$ **AllSkeleton**(i + 1, end)
>   **for each** $left$ **in** $leftTree$ **do**
>   **for each** $right$ **in** $rightTrees$ **do**
> $currentTree \leftarrow$ **new Node**($A[i]$)
> $currentTree.left \leftarrow left$
> $currentTree.right \leftarrow right$
>   **add** $currentTree$ **to** $allTrees$
>   **return** $allTree$

تعداد درخت ها را می توان با استفاده از رابطه بازگشتی به صورت زیر نوشت:

$$T(n) = \begin{cases} \sum_{i=1}^{n} T(i-1)T(n-i) & n \geq 1 \\ 1 & n = 0 \end{cases}$$

$$\sum_{i=1}^{n} T(i-1)T(n-i) = \sum_{i=0}^{n-1} T(i)T(n-i-1)$$

تابع مولد $B(x)$ را در نظر می گیریم:

$$B(x) = \sum_{n=0}^{\infty} T(n)x^n$$

$$= 1 + \sum_{n=1}^{\infty} T(n)x^n = 1 + \sum_{n=1}^{\infty}\sum_{i=0}^{n-1} T(i)T(n-i-1)x^n = 1 + x\sum_{n=1}^{\infty}\sum_{i=0}^{n-1} T(i)x^iT(n-i-1)x^{n-i-1}$$

$$= 1 + x\sum_{n=0}^{\infty}\sum_{i=0}^{n} T(i)x^iT(n-i)x^{n-i} = 1 + xB^2(x)$$

$$\Rightarrow B(x) = 1 + xB^2(x) \Rightarrow xB^2(x) - B(x) + 1 = 0 \Rightarrow B(x) = \frac{1}{2x}\left(1 \pm \sqrt{1-4x}\right)$$

برای این که $B(x)$ حول نقطه $x = 0$ خوش رفتار باشد $B(x) = \frac{1}{2x}\left(1 - \sqrt{1-4x}\right)$ را انتخاب می کنیم.

بسط دو جمله ای $\sqrt{1-4x}$ را می نویسیم:

$$\sqrt{1-4x} = \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4x)^n$$

$$B(x) = \frac{1}{2x}\left(1-\sqrt{1-4x}\right) = \frac{1}{2x}\left(1 - \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4x)^n\right) = -\frac{1}{2x}\sum_{n=1}^{\infty} \binom{\frac{1}{2}}{n} (-4x)^n$$

$$= -\frac{1}{2}\sum_{n=1}^{\infty} \binom{\frac{1}{2}}{n} (-1)^n (2)^{2n} (x)^{n-1}$$

$$\binom{\frac{1}{2}}{n} = \frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2)\dots(\frac{1}{2}-(n-1))}{n!}$$

$$= \frac{(-1)^{n-1}(1 \times 1 \times 3 \times 5 \times \cdots \times (2(n-1)-1))}{n!2^n}$$

$$= \frac{(-1)^{n-1}(1 \times 1 \times 3 \times 5 \times \cdots \times (2(n-1)-1))(2 \times 4 \times 6 \times \cdots \times (2(n-1)))}{n!2^n(2 \times 4 \times 6 \times \cdots \times (2(n-1)))}$$

$$= \frac{(-1)^{n-1}(2(n-1))!}{n!2^{2n-1}(n-1)!}$$

پس داریم:

$$B(n) = \sum_{n=1}^{\infty} \frac{(2(n-1))!}{(n-1)!n!} x^{n-1} = \sum_{n=0}^{\infty} \frac{(2n)!}{n!(n+1)!} x^n = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} x^n$$

چون $T(n)$ ضریب جمله x^n در تابع مولد است پس:

$$T(n) = \frac{1}{n+1}\binom{2n}{n}$$