091M4041H - Assignment 3
Algorithm Design and Analysis

Song Qige 2017E8018661044

2017 年 11 月 17 日

1 判断数字序列能否构成图(1)

1.1 algorithm describe and pseudo-code

将序列按照非升序排列。每次取第一个元素arr[i]，将后面的arr[i]个元素的值-1。如果i+arr[i]超过n，说明不能构成无向图。如果-1的过程中出现负值，也不能构成无向图。arr[i]个元素的值-1完成后重新对序列排序。如果i从1到n 走完后arr[n]为0，说明可以构成无向图。

```
1: function DecideGraph(d[], len)
2:     for i = 1 → len do
3:         sort(d+i,d+len)
4:         if i + d[i] > n then
5:             return false
6:         end if
7:         for j = i + 1 → len do
8:             d[j] − −;
9:             if d[i] < 0 then
10:                return false
```

```
            end if
        end for
    end for
    if d[len]! = 0 then
        return false
    end if
    return true
end function
```

1.2 greedy-choice property

贪心性质：由非负数组成的非增序列$s:d_1,d_2,...,d_n(n\geq 2,d_1\geq 1)$可以构成无向图,当仅当序列$s1:d_2-1,d_3-1,...,d_{d1+1}-1,d_{d1+2},....,d_n$是可以构成无向图的。

1.3 Prove the correctness

本解法采用了图论中判断可图性的Havel－Hakimi定理：由非负数组成的非增序列$s:d_1,d_2,...,d_n(n\geq 2,d_1\geq 1)$是可图的，当仅当序列$s1:d_2-1,d_3-1,...,d_{d1+1}-1,d_{d1+2},...,d_n$ 是可图的。序列s1 中有n-1 个非负数，s序列中d_1 后的前d_1 个度数减1后构成s_1中的前d_1个数。

定理证明：对于序列$s:d_1,d_2,...,d_n$，和序列$s1:d_2-1,d_3-1,...,d_{d1+1}-1,d_{d1+2},...,d_n$。如果s1 可图化，将s中最大度的点d1与s1中d1个点连一条边，可得到s图化的结果。

对于本题中的实现函数:

True Output:当函数返回值为真时，说明每次排序后的度序列中的最大值（设为d_j）都可以与度序列中剩余d_j个值代表的点连成边，直到所有的度均表现为边的形式，因此可以得到无向图G。

False Output:当函数返回值为假时，可能是排序后的度序列中的最大值d_j超过度序列中剩余值可以代表的点的个数，或度序列中的最大值d_j都不能与每一个与度序列中剩余的d_j个值代表的点连成边，这样该度数值无法表现为边，因此度序列无法构成无向图。

1.4 time complexity

排序的时间复杂度¡$O(n*(n*logn))$,总体时间复杂度$T(n) < O(n(n*logn+cn)) = O(n^2*logn)$

2 判断s是否为t的子串(3)

2.1 Algorithm Description and pseudo-code

分别使用i和j从头遍历s和t，如果当前位置s[i]=t[j]，将i与j同时后移，否则只后移j。如果i可以移动到s结束则s是t的子串，如果j已经到t结束i还未到s结束则s不是t的子串。

```
1: function ISSUBSEQUENCE(s[], s_len, t[], t_len)
2:     while i < s_len and j < t_len do
3:         if s[i] == t[j] then
4:             i++
5:             j++
6:         else
7:             j++
8:         end if
9:     end while
10:    if i == s_len then
11:        return true
12:    else
13:        return false
14:    end if
15: end function
```

2.2 greedy-choice property

贪心性质：将指针i,j分别从s和t的开头向后移动。如果s[i]=t[j]，同时后移指针i与j，否则后移j，继续判断s在i后面的序列是否为t在j后面的序列的子序列。

2.3 Prove the correctness

True Output:函数返回值为真说明s字符串的指针i从开始移动到了s结束，说明s的每个字符元素都在t中找到了相同元素，并且保持了原来的顺序。因此s是t的子串成立。
False Output:函数返回值为假说明当t字符串的指针j已经移动到t结束时，s字符串中仍然有剩余字符未在t中找到相同元素，因此s是t的子串不成立。

2.4 time complexity

设字符串t的长度为m，则时间复杂度为$O(m)$

3 Programming(5)

3.1 result analysis

Aesop_Fables.txt文件原大小为186KB,Huffman编码后的文件大小为103KB。压缩率为54.497%。graph.txt文件原大小为2198KB,Huffman 编码后的文件大小为910KB。压缩率为41.401%。（压缩文件时，将原文件经哈夫曼编码后得到的01串每8个存为1个字符到编码文件中。）
Aesop_Fables.txt文件加入个人代码信息编码（SongQige2017E8018661044）后解码结果:

```
A THRUSH was feeding on a myrtle-tree and did not move from it
because its berries were so delicious. A Fowler observed her
staying so long in one spot, and having well bird-limed his reeds,
caught her. The Thrush, being at the point of death, exclaimed, "O
foolish creature that I am! For the sake of a little pleasant food I
have deprived myself of my life."
The Rose and the Amaranth
AN AMARANTH planted in a garden near a Rose-Tree, thus
addressed it: "What a lovely flower is the Rose, a favorite alike
with Gods and with men. I envy you your beauty and your
perfume." The Rose replied, "I indeed, dear Amaranth, flourish but
for a brief season! If no cruel hand pluck me from my stem, yet I
must perish by an early doom. But thou art immortal and dost
never fade, but bloomest for ever in renewed youth."
The Frogs' Complaint Against the Sun
ONCE UPON A TIME, when the Sun announced his intention to
take a wife, the Frogs lifted up their voices in clamor to the sky.
Jupiter, disturbed by the noise of their croaking, inquired the
cause of their complaint. One of them said, "The Sun, now while
he is single, parches up the marsh, and compels us to die
miserably in our arid homes. What will be our future condition
if he should beget other suns?'
End
LastIndexSongQige2017E8018661044
```

graph.txt文件加入个人代码信息编码后解码结果：

```
19695 19730 9
19699 19809 17
19702 19762 13
19705 19732 14
19715 19964 18
19717 19997 13
19717 19894 14
19727 19815 18
19731 19755 10
19732 19986 18
19736 19834 5
19744 19902 2
19766 19943 11
19792 19924 3
19810 19841 16
19817 19825 6
19822 19835 20
19830 19857 14
19832 19965 18
19855 19912 19
19865 19907 3
19868 19967 7
19938 19973 3
19945 19947 11SongQige2017E8018661044
```