作业2,4,6

周天易

2022年12月26日

饮 水 思 源 • 爱 国 荣 校

作业四

6.17：

设b(i) 表示面值i是否能被找开，有：

$$\begin{cases} if\ (i<0):b(i)=false \\ if\ (i=0):b(i)=true \\ else:b(i)=\underset{1\le j\le n}{OR}\left(b\left(i-x_j\right)\right) \end{cases}$$

最多共有O(v)个状态，每次状态转移的代价为O(n)，于是整个算法的时间复杂度O(nv)

1.没考虑i<0的，不扣分

2.需要计算复杂度

6.20

6.20：

设c(i,j)为单词子集w[i,j]所构成的最优查找树的查找代价

最优查找树的子树也同样是最优查找树

$$c(i,j)=\min_{i\le k\le j}\left(c(i,k-1)+c(k+1,j)\right)+\sum_{i\le k\le j}p_k$$

c(i,i)=pi，c(i,i-1)=0

由下向上回溯，从1到n

1.复杂度，可以不计算，但应该n的3次方

2.pk没有求和不正确

3.相同方法不同表达也正确

6.21：

选取任意节点r的根，对应于下方悬挂的子树，每个节点可以定义一个子问题

I(u)=u下悬挂子树的最小覆盖集规模

最小覆盖是最大独立集的补集,即总顶点数减去I(root)

递推公式

$$I(u) = \max\{1 + \sum_{\text{grandchildren } w \text{ of } u} I(w), \sum_{\text{children } w \text{ of } u} I(w)\}$$

时间为O(|V|+|E|)，总状态为O(|V|)，状态转移总代价为O(|E|)

1.少数人推导了一遍

2.复杂度考虑不全面，不扣分

6.22题目

6.22. Give an $O(nt)$ algorithm for the following task.

Input: A list of n positive integers $a_1, a_2, \ldots, a_n$; a positive integer t.
Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once.)

(*Hint:* Look at subproblems of the form "does a subset of $\{a_1, a_2, \ldots, a_i\}$ add up to s?")

6.22

6.22：

设K(i,t)代表使用了前i个数能加和位t，则有

1.K(i,t)=K(i-1,t-ai)，使用ai加和

2.K(i,t)=K(i-1,t)，没有使用ai加和

K(i,t)=K(i-1,t-ai)VK(i-1,t)

填写一张nt的二维表格，每个单元填写时间为O(1)，总时间复杂度为O(nt)

1.二维表格\一维表格，都对

2.复杂度是O(nt)不是O(nv)，不扣分

3.没有给出逻辑或的式子，说清楚不扣分

4.有一些人没有用前i个数这个概念，解法不对

作业二

2.13：

a.图略

b.动态规划。考虑一颗n节点而且full的二叉树Tn，其左子树的节点数分别可能为1,3,5,..., n-2，而相应的右子树的节点数分别为n-2,n-4,...,1，由简单的排列组合关系可以得出下式：

$$B_n = B_1 B_{n-2} + B_3 B_{n-4} + B_5 B_{n-6} + \ldots + B_{n-2} B_1 = \sum_{i=1,3,5\ldots}^{n-2} B_i B_{(n-1-i)}$$

如果存在系列Cn满足C0 =1，且有递归式 $C_{n+1} = \sum_{k=0}^{n} C_k \cdot C_{n-k}$ ，与上面 Bn的关系式相比较，可以看出，$B_{2n+1} = C_n$ ，这个Cn 即是著名的 Catalan 数，另有通项公式$C_n = \frac{1}{n+1}\binom{2n}{n}$

以及递推式 $C_n = \frac{4n-2}{n+1} C_{n-1}$

2.19：

a) 时间复杂度：

$$T(k,n)=O(n+n)+O(2n+n)+O(3n+n)+\cdots+O((k-1)n+n)$$
$$=O(2n+3n+\cdots kn)=O(k^2n)$$

b) 分治，先分别 *merge* 前 $\left\lfloor \frac{k}{2} \right\rfloor$ 个序列和后 $\left\lceil \frac{k}{2} \right\rceil$ 个序列，再将中间结果组合起来：

$$\begin{cases} T(k,n)=T(\lfloor k/2 \rfloor,n)+T(\lceil k/2 \rceil,n)+O(kn) \\ T(1,n)=O(1) \end{cases}$$
$$\Rightarrow T(k,n)=O(k\log kn)$$

1.极少数计算错误

2.22

2.22：

设两个序列分别为 A[1,...,m]， B[1,..., n]， 首先比较A[m/2]和B[n/2]，再根据比较结果进行判断，不失一般性，我们假设A[m/2]≥B[n/2]，再由，AB均有序可知，A后一半的所有元素肯定排在A∪B的(m+n)/2名之后，而B前一半的所有元素肯定排在A∪B的(m+n)/2名以内。所以，如果k≤(m+n)/2，就可以将A后一半的所有元素剔除掉，继续在剩余元素中寻找第k小元素，反之同理。于是可得以下递归式：

$$\begin{cases} T(m,n)=T(m/2,n)+1 & \textit{if cond...} \\ T(m,n)=T(m,n/2)+1 & \textit{else} \\ T(0,n)=T(m,0)=O(1) & \end{cases}$$

T(m,n)=O(logm+logn)

2.28

2.28：

$$H_k v = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix} \begin{bmatrix} v_u \\ v_d \end{bmatrix} = \begin{bmatrix} H_{k-1}(v_u + v_d) \\ H_{k-1}(v_u - v_d) \end{bmatrix}$$

$$\Rightarrow T(n) = 2T(n/2) + O(n) \Rightarrow T(n) = O(n \log n)$$

作业六

8.3：

1.解中变量为true的个数不大于k，再将解带入判断结果是否为true，可在多项式时间内验证的，属于NP。

2.可以将SAT归约到STINGY SAT（将k设为所有变量的总个数即可），于是可知STINGY SAT为NP完全问题。

3.SAT是NP完全问题，所以STINGY SAT是NP完全问题

A是NP完全问题的解法：

1.A是NP问题

2.NPC问题B可以规约到A

8.9

8.9：

1.证明HITTING SET问题是 NP 问题。对于HITTINGSET问题的每个解 H , 只需判断对所有 i 是否满足H和Si不相交，且H规模小于b，可在多项式时间内验证。

2.很容易将最小顶点覆盖归约到 HITTING SET。假设要求图G的最小顶点覆盖，可以建立一个HITTING SET实例，其中边集合为{S1,S2,...,Sn}，点集合为{v1,v2,..,vn}，问题变为大小不超过b的点集H，边集合中每个元素Si都至少有一个元素在H中，这就是图G的最小顶点覆盖。

3.最小顶点覆盖是NP完全问题，所以HITTING SET是NP完全问题

8.14

8.14：

1.对于给定的无向图G，整数k，任意给出一个可能解H，可在多项式时间内验证H是否为规模k的团或者独立集

2.可以将最大团问题归约到此问题。假设要求任意图G(V,E)中大小为k的团，可以在图G中添加k个相互独立的顶点，得到新图G '，有规模为k的独立集。这新加的k个顶点保证了图G '存在大小为k的独立集，同时又不影响到原图的团。

3.最大团问题是NP完全问题，所以该问题是NP完全问题。

8.19

8.19：

类似于8.14。可以将团问题归约到KITE问题。若要求图G(V,E)的最大团，可以在图G中添加|V|个新顶点，并将每个新顶点都连向原图中不同的某个顶点，共形成了|V|条新边，这样就得了一个新图G '。容易看出，在G '中存在大小为2g的kite当且仅当G中存在大小为g的团。