算法基础HW6

PB19071535徐昊天

EX1

答：

(a)

根据题中所示切比雪夫的素数分布定理：

存在$C_1, C_2 = O(1)$使得$\pi(n)$属于$[C_1\frac{n}{logn}, C_2\frac{n}{logn}]$。

故根据定义$\pi(n)$为小于等于n的质数个数，即$[n, Cn]$中质数个数为$\pi(Cn) - \pi(n-1)$。

有：$\pi(Cn) - \pi(n-1) \geq \pi(Cn) - \pi(n) \geq C_1\frac{Cn}{logCn} - C_2\frac{n}{logn}$

只需证$C_1\frac{Cn}{logCn} - C_2\frac{n}{logn} \geq 1$

即：$C \geq \frac{logCn}{C_1n} + \frac{C_2logCn}{C_1logn} = \frac{logCn}{C_1n} + \frac{C_2(logC+logn)}{C_1logn}$

考虑n趋于无穷大时，上式右边趋近于$\frac{C_2}{C_1}$，故取$C \geq \frac{C_2}{C_1} + 1$，显然以上不等式能够成立，故必存在常数$C$使得区间$[n, Cn]$中至少包含一个质数。

综上，原命题得证。

(b)

(1)

已知561=3×11×17，故561不为质数。

由于3、11、17皆为质数，根据费马小定理有：

若a满足$(a, 3) = 1$，则$a^2 \equiv 1(mod \quad 3)$
若a满足$(a, 11) = 1$,则$a^{10} \equiv 1(mod \quad 11)$
若a满足$(a, 17) = 1$,则$a^{16} \equiv 1(mod \quad 17)$
由于3、11、17皆为质数，若a满足$(a, 3) = 1$，$(a, 11) = 1$，$(a, 17) = 1$，则满足$(a, 561) = 1$
即已知$(a, 561) = 1$

故有：$3|(a^2-1), 11|(a^{10}-1), 17|(a^{16}-1)$。

由于已知:

$$(a^{560}-1) = (a^2-1)(a^{558} + a^{556} + \ldots + a^2 + 1)$$
$$(a^{560}-1) = (a^{10}-1)(a^{550} + a^{540} + \ldots + a^{10} + 1)$$
$$(a^{560}-1) = (a^{16}-1)(a^{544} + a^{528} + \ldots + a^{16} + 1)$$

故有$(a^2-1)|(a^{560}-1), (a^{10}-1)|(a^{560}-1), (a^{16}-1)|(a^{560}-1)$，根据整除的传递性有：
$3|(a^{560}-1), 11|(a^{560}-1), 17|(a^{560}-1)$。

由于3、11、17皆为质数，故有：

$3 * 11 * 17 = 561|(a^{560}-1)$。

即可化简为$a^{560} \equiv 1(mod \quad 561)$，又已知$(a, 561) = 1$，即561满足费马小定理，故561为伪质数。

(2)

令$n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$(其中$p_1, p_2 \dots p_r$皆为互不相等的质数)，显然任何正整数都可用此式表示，则$a \in \{1, 2, \dots, n-1\}$满足$(a, n) = 1$的个数为$p_1^{e_1-1} p_2^{e_2-1} \cdots p_r^{e_r-1}(p_1 - 1)(p_2 - 1) \cdots (p_r - 1)$。

理由如下：

令$a \in \{1, 2, \dots, n-1\}$满足$(a, n) = 1$的个数为$g(n)$。参考**代数结构**课本中的欧拉函数一节：

首先证：**若p为素数，则对一切正整数n，**$g(p^n) = p^{n-1}(p-1)$

证明如下：

显然小于 p^n 的数共有 $p^n - 1$ 个。其中与 p^n 有公因子 p 的数是$p, 2p, \cdots, (p^{n-1} - 1)p$，一共有 $p^{n-1} - 1$ 个。那么与 p^n 无公因子p的，即与p^n互素的数共有：
$(p^n - 1) - (p^{n-1} - 1) = p^{n-1}(p-1)$个。

故以上命题得证。

再证：**当(m,n)=1时，g(mn)=g(m)g(n)。**

证明如下：

g(mn)为小于mn且与mn互素的正整数个数，以下把所有小于等于mn的正整数列成一个方阵：

$$\begin{pmatrix} 1 & m+1 & 2m+1 & \cdots & (n-1)m+1 \\ 2 & m+2 & 2m+2 & \cdots & (n-1)m+2 \\ 3 & m+3 & 2m+3 & \cdots & (n-1)m+3 \\ \vdots & \vdots & \vdots & & \vdots \\ m & m+m & 2m+m & \cdots & (n-1)m+m \end{pmatrix}$$

若 $(m, r) = d > 1$，那么 r 所在行的全部元素 $r, m+r, 2m+r, \cdots, (n-1)m+r$ 均与 mn 有公因子 d。由此可知，与 mn 互素的数只能在 $(m, r) = 1$ 的 $g(m)$ 行中寻找，而当 $(m, r) = 1$ 时，$\{r, m+r, 2m+r, \cdots, (n-1)m+r\}$ 是 n 元集合，并且两两模 n 不同余。它是模n的完系。在一个模n的完系中有 $g(n)$ 个数与 n 互素。而该完系中每个数均与 m 互素，从而它里面有$g(n)$个数与mn互素。

故以上命题得证。

根据以上得出的两条结论即可推导出原问题的解：

$$\begin{aligned} g(n) &= g(p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}) \\ &= g(p_1^{e_1}) g(p_2^{e_2}) \dots g(p_r^{e_r}) \quad (\text{由于} p_1, p_2 \dots p_r \text{互质，故} p_1^{e_1}, p_2^{e_2} \dots p_r^{e_r} \text{也互质}) \\ &= p_1^{e_1-1}(p_1 - 1) p_2^{e_2-1}(p_2 - 1) \dots p_r^{e_r-1}(p_r - 1) \\ &= p_1^{e_1-1} p_2^{e_2-1} \dots p_r^{e_r-1}(p_1 - 1)(p_2 - 1) \dots (p_r - 1) \end{aligned}$$

以上即为题中所求结论。

EX2

答：

首先在二分图中查找最大匹配，参考算法导论中的内容，在二分图中构建一个流网络G'(V'，E')，其中的流对应于匹配。设流网络中源节点s和汇点t为不属于G的新结点，并设V'=V∪{s，t}。由于G中结点集划分为V=L∪R，故E中从L指向R的边都是流网络G'的边。G'中的边定义如下：
$E' = \{(s,u) : u \in L\} \cup \{(u,v) : (u,v) \in E\} \cup \{(v,t) : v \in R\}$。完成流网络的创建并为E'中的每条边赋单位流量。调用Ford-Fulkerson函数查找流网络中的最大流，根据最大流即可得到最大匹配。

根据图论课本的$Konig - Egervary$定理，二分图中最大匹配数等于最小覆盖数。若L与R中的顶点全在最大匹配中，则L或R即为最小顶点覆盖；否则，假设L中有顶点不在最大匹配中，则令X为L中不在最大匹配中的顶点集合，令Z为和X中某个结点存在交错轨道的结点集合，S=L∩Z，T=R∩Z，则(L-S)∪T即为最小覆盖；假设R中有顶点同上。

对应伪代码如下：

```
function vertex_cover(G)
    let G'(V',E') be a new graph
    G'.V' = G.L∪G.R∪s∪t
    G'.E' = {(s,u):u∈L}∪{(u,v):(u,v)∈E}∪{(v,t):v∈R}
    for edge in G'
        edge.c = 1  // c储存剩余容量，剩余容量初始赋单位值
    Ford-Fulkerson(G',s,t)
    X = Y = M = S = T = ∅
    for edge in G
        if edge.c = 0
           M = M∪edge
        else
           // X存储L中未匹配的顶点，Y存储R中未匹配的顶点
           X = X∪edge.u, Y = Y∪edge.v
    if X == NIL && Y == NIL
        if |L| >|R|
            return R
        else
            return L
     else if X != NIL
        for v in L
            if v has a staggered track with any vertex in X
                 S = S∪v
        for v in R
            if v has a staggered track with any vertex in X
                 T = T∪v
        return (L-S)∪T
      else
        for v in L
            if v has a staggered track with any vertex in Y
                 S = S∪v
        for v in R
            if v has a staggered track with any vertex in Y
                 T = T∪v
        return (R-T)∪S
```

时间复杂度分析：

由于|V'|=|V|+2=|L|+|R|+2，故|V'|=O(V);由于结点集V中的每个结点至少有一个相连的边，|E|≥|V|/2。因此|E|≤|E'|=|E|+|V|≤3|E|，所以|E'|=O(E)。

对算法的每一个部分分别分析时间，该算法中构造G'部分时间复杂度为O(V'+E')=O(V+E)，调用Ford-Fulkerson算法时间复杂度为O(V'E')=O(VE)，处理最大流并查找最大匹配的部分时间复杂度为O(E)，最终返回最小覆盖的部分时间复杂度为O(V)。

综上，时间复杂度为O(VE)。

正确性分析：

根据**图论**课本的$Konig-Egervary$定理，二分图中最大匹配数等于最小覆盖数。分以下几种情况考虑：

①L与R中的顶点全在最大匹配中

L与R中顶点较少的一个集合即为所求最小顶点覆盖。该集合可覆盖二分图中所有顶点且若减少任何一个顶点即不可实现覆盖，故该结论显然正确。

②存在L或R中的顶点不在最大匹配中

根据对称性，假设L中有顶点不在最大匹配中(R中有顶点不在最大匹配中同理)，则令X为L中不在最大匹配中的顶点集合，令Z为和X中某个结点存在交错轨道的结点集合，S=L∩Z，T=R∩Z，则B=(L-S)∪T即为最小覆盖。用反证法，假设B不是最小覆盖，则存在一条两个端点分属 S 和 R-T的边，这与 N(S)=T (T是S的邻接结点集合)矛盾。另一方面，由于 M 是最大匹配，G 中没有关于 M 的可增广轨道，所以 T 中所有的顶点都被 M 匹配；又因为 X 中所有没有被 M 匹配的顶点都属于 $U \subseteq S$，所以 X-S 中顶点与 U 中顶点没有交错轨道相连，所以 X-S 中顶点不可能与 T 中顶点相配。所以|M|=|(X-S)∪T|=|B|。故可证 B 是最小覆盖。

综上可证，该算法正确。

EX3

答：

(a)

线性规划的目标为：

$$minimize \sum_{v \in V} x_i \quad i = 1..n$$

线性规划的约束为：

$$\begin{aligned} x_u + x_v \geq 1 \quad (u,v) \in E \\ x_v \geq 0 \quad v \in V \\ x_v \leq 1 \quad v \in V \end{aligned}$$

(b)

证明 S 是一个顶点覆盖:

考虑任意边(u，v)∈E，根据约束$x_u + x_v \geq 1$，可知：x_u、x_v至少有一个的值大于等于1/2，根据S的定义，u和v至少有一个将存在于S中。故图G中的每一条边都将被覆盖，可证S是一个顶点覆盖。

证明 S 同最小的顶点覆盖相比的近似比不大于2:

令C^*是顶点覆盖问题的一个最优解，z^*是以上线性规划的一个最优解。由于最优的顶点覆盖也是该线性规划的可行解，故z^*必定是C^*的一个下界。即：$z^* \leq C^*$。

考虑顶点覆盖：$z^* = \sum_{v \in V} x_v \geq \sum_{v \in V, x_i \geq 1/2} x_v \geq \frac{1}{2} \sum_{v \in V, x_i \geq 1/2} \geq = \frac{1}{2} \sum_{v \in S} = \frac{1}{2} C$

故有：$C \leq 2z^* \leq 2C^*$，即可证 S 同最小的顶点覆盖相比的近似比不大于2。

EX4

答：

证明如下：

①证明：$EQ-SUM \in NP$

已知n个正整数$s_1, s_2, \dots s_n$，给定证书P_1，P_2。显然可在多项式时间内验证是否满足$\sum_{s_i \in P_1} s_i = \sum_{s_j \in P_2} s_j$。故可证EQ-SUM∈NP。

②证明：$SUBSET-SUM \leq_p EQ-SUM$

SUBSET-SUM定义如下：

给定一个正整数的有限集 S 和一个目标数 t>0，找到一个子集 S′ ⊆ S 使得 S′ 的元素和为t。

令有限集S的元素和为s，设X=S∪{s-2t}为EQ-SUM的输入集合。显然归约算法在多项式时间以内。

再证：**S,t对SUBSET-SUM是可满足的当且仅当X对EQ-SUM是可满足的。**

=>：

由于S中存在子集S'使得S'中元素和为t，故S''=S-S'中元素和为s-t，S'''=S'∪{s-2t}中元素和也为s-t，则存在集合$P_1 = S'', P_2 = S'''$且$X = P_1 + P_2$，并满足$\sum_{s_i \in P_1} s_i = \sum_{s_j \in P_2} s_j$，即X对EQ-SUM是可满足的。

<=：

由于X分为两部分P_1, P_2，且两个集合的元素和皆为s-t。由于元素s-2t存在于其中一个部分，故此集合除去s-2t得到的子集S'中元素之和为(s-t)-(s-2t)=t，且S'中元素都属于集合S。即S对SUBSET-SUM是可满足的。

综上，原命题得证。

EX5

答：

(a)染色问题是NP难问题。证明如下：

将该优化问题重新表述为判定问题：

k-COLOR={<G,k>:图G可以使用最多k种颜色染色}。

通过证明此判定问题为 NP-complete 来得到染色问题 NP 难的结论。

首先证明3-染色是NPC问题：

①证明：$3-COLOR \in NP$

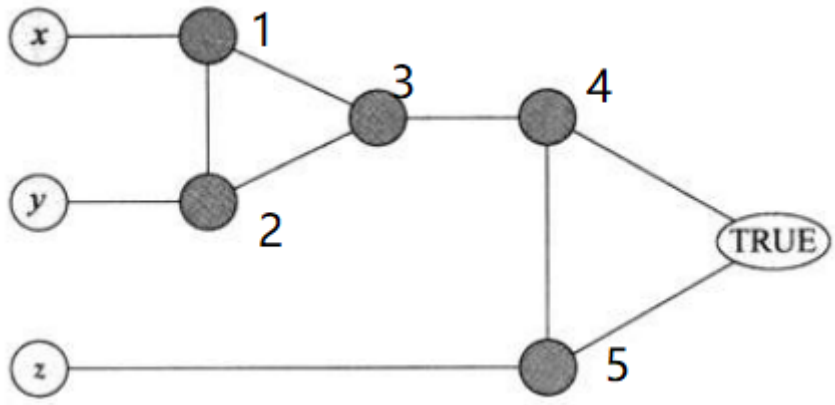
假定已知图G，给定染色方案G'，可直接通过一次遍历搜索检查G'是否用至多3种颜色进行染色并保证相邻顶点的颜色都不同，可在多项式时间之内验证这一问题。故可证COLOR∈NP。

②证明：$3-SAT \leq_p 3-COLOR$

给定变量$x_1, x_2, \dots, x_n$上的一个3-CNF公式，它由m个子句构成。通过以下方式从公式构造图形：

- 对于每个变量x_i在图中构造一个顶点v_i和一个顶点v_i'表示变量x_i的否定。
- 额外添加了三个不同颜色的顶点，分别表示 True、False 和 Red的值。
- 在添加的顶点 True、False 和 Red之间添加边以形成三角形。

- 在顶点v_i和v_i'和Red点之间添加边以形成三角形。
- 对于每个子句x V y V z，添加5个顶点，并把子句中的文字与特殊顶点True相连。如下图所示：



存在以下约束：

1. 对于顶点对v_i和v_i'，其中一个被分配为TRUE，另一个被分配为FALSE。
2. 对于 m 个子句中的每个子句 c，至少有一个文字为 TRUE 值才能使该值为真。

再证：**3-SAT是可满足的当且仅当3-COLOR是可满足的。**

=>:

假设 3-SAT 公式是可满足的，可知在每个子句中，x、y、z至少有一个被着色为TRUE，因此，对应的v_i可分配TRUE，而v_i'可分配FALSE。将x、y、z所有着色可能列出下表(根据对称性，x,y存在对称关系的情况在表中省略)：

x	y	z	1	2	3	4	5
TRUE	TRUE	TRUE	FALSE	RED	TRUE	RED	FALSE
TRUE	FALSE	TRUE	FALSE	TRUE	RED	FALSE	RED
TRUE	TRUE	FALSE	FALSE	RED	TRUE	FALSE	RED
TRUE	FALSE	FALSE	FALSE	RED	TRUE	FALSE	RED
FALSE	FALSE	TRUE	TRUE	RED	FALSE	RED	FALSE

由上可知，所有情况下上图都可满足3-COLOR。

<=:

若存在子句C_j = (x V y V z)，使得 x、y、z 都是 False。则对应上图中结点1和结点2需分别为TRUE和RED（或相反），即结点3必须为False,由于结点3和结点z皆为False，故结点4和5之间必然存在True，这与3-COLOR存在矛盾，因为会有邻接的TRUE结点。故3-SAT是不可满足时3-COLOR也是不可满足的，根据逆否命题的性质，可证3-COLOR可满足可推出3-SAT可满足。

综上，可证3-染色是NPC问题。根据以上证明过程，可引申证明k-COLOR是NPC问题。

由于可通过多次调用判定问题在多项式时间内归约到染色问题的优化问题，故可证染色问题是NP难问题。

(b)染色问题不是NPC问题。理由如下：

由于已知染色问题是NP hard问题，要证染色问题不是NPC问题，只需证染色问题$\notin$NP。

假定已知图G和整数k，要证明k是对G染色的最少颜色数量。故需要利用(a)问中的判定问题分别判断图是否能被k种颜色染色与是否能被k-1种颜色染色。由于已知(a)问中的判定问题是一个NPC问题，故显然以上证明过程不可在多项式时间内完成。故可证染色问题$\notin$NP。

综上可证，染色问题不是NPC问题。

(c)

任取一个起始顶点并任意赋予一种颜色，通过BFS对领接顶点赋予另一种颜色，若中途发现存在邻接顶点已经被赋予和自己相同的颜色，则返回不存在2-染色实例。

```
function BFS(G)
    let s be a random vertex in G.V
    for each vertex u ∈ G.V-{s}
        u.color=WHITE   // 初始颜色，WHITE为未染色时的颜色
    s.color=color1
    Q=∅
    ENQUEUE(Q,s)
    while Q!=∅
        u=DEQUEUE(Q)
        for each v ∈G.Adj[u]
            if v.color == WHITE
                let v.color be the different color of u
                ENQUEUE(Q,v)
            else if v.color is same as u.color
                return FALSE
    return TURE
```

时间复杂度证明：

由于该算法是基于BFS修改后的算法，对于每个顶点和每条边至多遍历一次，故时间复杂度为$O(V+E)$。

正确性证明：

由于2染色问题的目的是用两种颜色对图中所有顶点进行染色，故BFS的起始顶点的选择不会对染色实例的存在与否产生影响，因为该顶点迟早会在搜索过程中被遍历到。且由于两种颜色的对称性，起始节点的颜色选择对染色实例的存在与否也不会产生影响。故起始顶点和起始顶点的颜色都不会对染色实例的存在与否产生影响。

若染色实例存在，则由于BFS过程中对顶点染色时确保了与邻接顶点颜色不同，故能够得到符合相邻顶点的颜色都不同的染色实例。

综上可证，该算法正确。