Solution7

1.上课介绍了3DM（NP-C问题）。请设计一个算法使得能够在多项式时间复杂度实现2DM（P问题）。如给定一个二分图$G=(U,V,E)$，其中

- U和V是两个不相交、内部不相连的顶点集，且有$|U|=|V|=N$；
- E是连接U和V之间的边的集合。

若存在完美匹配方案，使得U中的N个顶点能与V中的N个顶点一一匹配，则你给出的算法需要在多项式时间复杂度内回答“Y”，并返回一个（可能有多个）完美匹配方案；若给定的图中不包含完美匹配，算法也需要在多项式时间内回答“N”。

考虑匈牙利算法：

1. 初始化:创建一个匹配状态数组 `match`，用于记录每个顶点的匹配情况。初始时，所有顶点均未匹配。
2. 增广路径搜索:对于U中的每个顶点 u，尝试找到一条增广路径，使得 u 可以与 V 中的某个未匹配顶点 v 进行匹配。使用深度优先搜索（DFS）来实现该过程：
    - 对于u的所有邻接顶点 v
        - 如果 v 尚未被访问，标记 v 为已访问。
        - 检查 v 是否未匹配，或者 v 的当前匹配对象能通过递归找到一个新的增广路径。如果可以找到，则将 u 和 v 进行匹配。
3. 重复搜索:重复以上过程，直到所有可能的增广路径都被尝试。
4. 结果输出:如果匹配的大小等于N，则存在完美匹配，输出“Y”并返回匹配方案。否则，输出“N”。

```
bool find(int u) {
    for (int v : G[u])
        if (!st[v]) {
            st[v] = 1;
            if (match[v] == 0 || find(match[v])) {
                match[v] = u;
                return 1;
            }
        }
    return 0;
}
```

2.上课介绍了 $Road\ Cutting$ 问题的一种做法，即 $Relation$为$x(l)=\max\{v(p)+x(l-p)|p\in\{1,\dots,l\}\}$。现改为$x(l)=\max\{x(p)+x(l-p),v(l)|p\in\{1,\dots,l-1\}$，表示长度为$l$的木条最大收益 $x(l)$ 是从所有可能的切割位置 p 中选择一个，使得当前长度为 p 的部分收益 $x(p)$ 加上剩余长度 $l-p$ 的 $x(l-p)$ 的和最大，或者直接使用整段木条的价值 $v(l)$，从中选择最大的值。请分析修改后的算法的时间复杂度?

共l个不同的子问题，每个子问题暴力枚举l种切分，所以时间复杂度仍为$O(l^2)$

3.画出课上介绍的 $Subset\ Sum$ 问题状态转移对应的计算有向无环图（$Computation DAG$）。

$Subset\ Sum$问题：给定一个由 n 个正整数构成的集合 $S=\{a_0,a_1,\dots,a_{n-1}\}$ 和一个目标和 T。问**是否存在** S 的一个子集，使得该子集的元素之和正好等于 T。

$Relation$: $x(i,t)=x[i+1][t]$ or $x[i+1][t-a_i]$ if $t\geq a_i$

每个节点$x(i,t)$有两条出边：

- 一条指向 $x(i+1,t)$：表示不选取元素a_i。
- 另一条边指向$x(i+1,t-a_i)$：表示选取元素a_i（前提：$t\geq a_i$）。

图略。

4.以下是一个无法被满足的3SAT问题实例：

$$\Phi=(x_1\vee\neg x_2\vee x_3)\wedge(\neg x_1\vee x_2\vee x_3)\wedge(x_1\vee x_2\vee\neg x_3)\wedge(\neg x_1\vee\neg x_2\vee\neg x_3)\wedge(\neg x_1\vee\neg x_2\vee x_3)\wedge(x_1\vee\neg x_2\vee\neg x_3)$$

画出对应的3DM的实例。

有3个变量，6个子句。

$Variable\ Gadget$：对每一个变量构造12个“花瓣”（对应6个子句的$x[i]$和$\neg x[i]$）。

$Clause\ Gadget$：6个，每一个子句对应一个。

$Garbage\ Collector\ Gadget$：有24个内部点（12个$g_1[k]$和$g_2[k]$），36个外部点（$x_i[j]$和$\neg x_i[j]$，i代表3个变量，j代表6个子句）。

图略。