1

(1)

可通过归约停机问题来完成。对任意停机问题实例 (N, w)，构造新程序 M，工作如下：
对于任意输入 x：

1. 先模拟运行程序 N 并输入 w；
2. 若 $N(w)$ 停机，则 $M(x)$ 输出空并停机；
3. 若 $N(w)$ 不停机，则 $M(x)$ 输出非空值（如1）并停机。

此时，$N(w)$ 停机当且仅当 M 属于EMPTY。若存在算法可判定EMPTY，则该算法可直接用于判定停机问题，与停机问题的不可判定性矛盾。

(2)

类似构造，取 M_1 恒输出 1，M_2 当 $N(w)$ 停机时输出 1 否则输出 0 即可。

2

(1)

我们需要证明 $3-SAT\ is\ satisfiable \Leftrightarrow value(max-2-SAT)=7m$

当 $x=y=z=0$ 时，最多只有 x, y, z，$x \vee \neg w_i, y \vee \neg w_i, z \vee \neg w_i$ 6 个 clause 能够取值为 1。

当 $x=1, y=z=0$ 时，可以取 $w_i=0$ 使得 $x, \neg x \vee \neg y, \neg x \vee \neg z, \neg y \vee \neg z, x \vee \neg w_i, y \vee \neg w_i, z \vee \neg w_i$ 7 个 clause 能够取值为 1。

当 $x=y=1, z=0$ 时，可以取 $w_i=0$ 使得 $x, y, \neg x \vee \neg z, \neg y \vee \neg z, x \vee \neg \vec{w}_i, y \vee \neg w_i, z \vee \neg w_i$ 7 个 clause 能够取值为 1。

当 $x=y=z=1$ 时，可以取 $w_i=1$ 使得 $x, y, z, x \vee \neg w_i, y \vee \neg w_i, z \vee \neg w_i, w_i$ 7 个 clause 能够取值为 1。

其他情况同理。

(2)

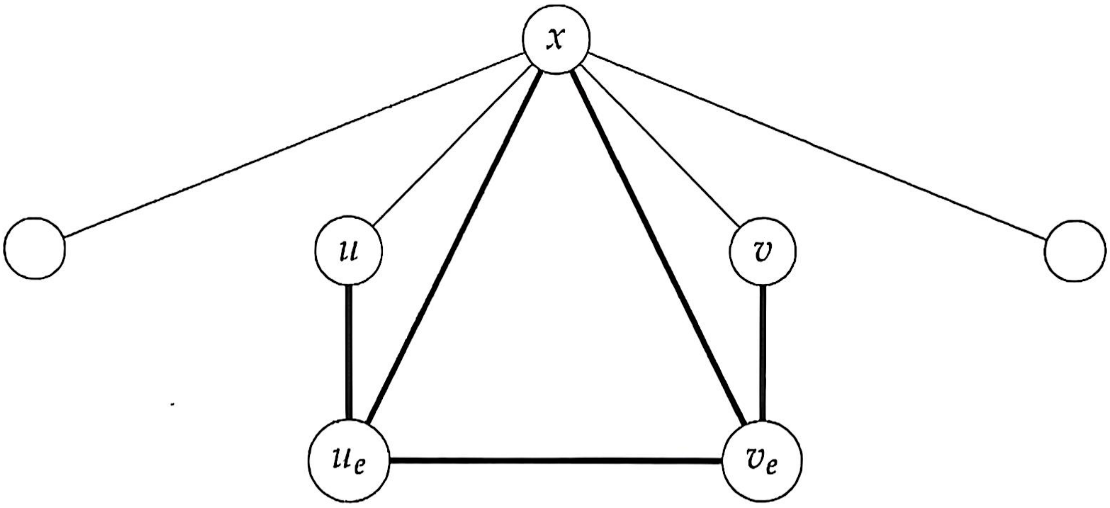
该问题属于 NP 易得，接下来证该问题属于 NP-hard，我们通过将 max-2sat 归约为该问题来说明这点。

首先可以通过 $O(\log m)$ 的二分，使得求最大满足的 clause 数量转化为是否存在满足大于等于 k 个 clause 数量的赋值。

我们参考课件 45 页构建 gadget 的方法。建立新点 y，并使用 y 来构建所需的约束。

对于每个变量 x_i，连边 $x_i - \neg x_i - y - x_i$，边权分别为 $M, 0, 0$，大数 M 保证最大割中每个变量只能取一个值。我们将与 y 处于同一集合的点视为处于 S 中。

对于每个 clause (u,v)，将 gadget 建立为如课件 45 页所示（其中 u_e 应为 $\neg u$ 而非新建虚拟节点，且不需要图中源点和汇点），我们可以通过保证 u,v 两个变量均被取为 $\neg u,\neg v$ 时割的容量最小，从而保证在求最大割的时候该 clause 尽可能被满足。例如令 $(x,u)=(x,v)=0$，其余边权均为 1 即可。此时当有一个变量满足或两个变量满足时割为 4，两个变量均不满足割为 2，所以最大割会尽量保证满足尽可能多的约束。



于是该图的最大割即为 $nM+4|\text{满足约束}|+2|\text{不满足约束}|$，故可以将 max-2sat 通过二分转化为判断是否存在割大于等于 k 的问题。

(3)

该条件等价于图是否为二分图，任意取一点开始 dfs，奇数层点染为黑色偶数层点染为白色，只要不存在边连接相同颜色的点即为二分图。