1. 在这个问题中你考虑下述顶点覆盖问题的简单随机算法。

开始时 $S=\emptyset$
While S 不是一个顶点覆盖
    选择一条没有被 S 覆盖的边 e
    随机地选择 e 的一个端点（每一个端点的可能性相等）
    把选中的结点加入 S
Endwhile

我们感兴趣的是这个算法选择的顶点覆盖的期望代价。

(a) 这个算法是**最小权顶点覆盖问题**的 c-近似算法吗？其中 c 是某个常数。证明你的答案。

(b) 这个算法是**最小规模顶点覆盖问题**的 c-近似算法吗？其中 c 是某个常数。证明你的答案。

（提示: 用P_e表示在这个算法中边 e 作为未覆盖的边被选的概率。你能够用这些概率表示解的期望值吗? 为了用这些概率给出最优值的上界，尝试给出所有与给定结点v关联的边的概率之和的上界，即$\sum_{e\,与v关联} P_e$的上界）

2. 并行和分布式系统的**负均衡算法**试图把一组计算任务分散到多台机器上。用这种方式，没有一台机器成为“热点”。如果能够进行某种集中的协调，那么负载可能被分散得近乎理想。如果任务来自各种不能协调的来源，那怎么办? 正如我们在 13.10 节（参考教材）一中看到的，种可能的做法是把它们随机地分配给机器，并希望这种随机化能防止不均衡。显然，一般地这不能做得像集中解决那样理想，但它可能是相当有效的。现在我们打算分析 13.10 节（参考教材）中考虑的一种简单的负载均衡启发式算法的几个变形与推广。

假设有 k 台机器和 k 项要处理的任务。每一项任务被独立地随机分配给一台机器（每台机器的可能性相等）。

(a) 设 N(k)表示没有接受到任务的机器的期望数，因而 N(k)/k 是无事可做的机器的期望比例。 极限$\lim\limits_{k\to\infty} N(k)/k$是多少? 证明你的答案。

(b) 假设机器不能让剩余的任务排队等候，因而随机分配把一件以上的任务送给机器 M，那么 M 将做它接受到的第一项任务并拒绝其余的任务。设 R(k) 是被拒绝的任务的期望数，因而 R(k)/k 是被拒绝的任务的期望比例。极限 $\lim\limits_{k\to\infty} R(k)/k$ 是多少? 证明你的答案。

(c) 现在假设机器有稍微大一点的缓冲器，每一台机器能做它接受的前两项任务并拒绝其他更多的任务。设$R_2(k)$表示在这个规则下被拒绝的任务的期望数。 极限 $\lim\limits_{k\to\infty} R_2(k)/k$ 是多少? 证明你的答案。

3. 在 13.4 节（参考教材）我们设计了 MAX 3-SAT 题的一个近到 7/8 因子的近似算法,

在那里我们假设每一个子句有与 3 个不相关联的项。现在要考虑一个类似的 MAX SAT 问题，给定变量集 $X = \{x_1, \dots, x_n\}$ 上的一组子集 $C_1, \dots, C_k$，求一个真值赋值满足尽可能多的子句。 每一个子句至少有一个项，一个子句中的所有变量是不同的, 除此之外对子句的长度没有做任何假设，可能有一些子句有很多变量，而另一些可能只有一个变量。

(a) 首先考虑我们用于 MAX 3-SAT 的随近算法, 以概率 1/2 独立地一个量为**真**或**假**。证明这个随机赋值满足的期望子句数不小于 k/2, 即所有的子句中至少期望地有一半被满足。给出一个例子说明存在 MAX SAT 实例使得任何赋值满足的子句都不超过所有子句的一半。

(b) 如果有一个只由一项成的子句（例如，仅由x_1，$\bar{x}_2$构成的子句）， 那只有唯一的方法满足它: 必须以适当的方式给对应的变量赋值。如果有两个子句, 其中一个仅由x_i构成，而另一个仅由它的否定项 $\bar{x}_i$ 构成，那么这是一个相当直接的矛盾。
假设实例不含这种"冲突的子句"对，即没有变量x_i使得有一个子句 C=$\{x_i\}$, 又有一个子句 C'=$\{\bar{x}_i\}$, 修改上面的随机算法把近似因子从 1/2 改进到 0.6, 即修改算法使得被满足的期望子句数至少为 0.6k。

(c) 试给一个一般 MAX SAT 题的多项式时间随机算法, 使得算法满足的期望子句数不少于最大可能的 0.6。

（注意, 根据部分(a)中的例子, 存在不可能满足多于 k/2 个子句的实例，这里的要点是仍可指望得到一个有效的算法, 它能够期望地满足最优赋值能够满足的最大子句数的 0.6。）