Ch8

1.算法 Approximate_Vector_Cover(G)

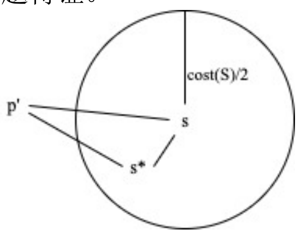
输入 :无向图 G=(V, E)

输出 :C⊆V，满足:(1)∀(u, v)∈E，u∈C 或 v∈C; (2)C 是满足(1)的最小集合

1:	C=0
2:	E'=E[G]
3:	**While** E'≠0 **DO**
4:	任取(u, v)∈E'
5:	C=C∪{u, v}
6:	从 E'中删除所有与 u 或 v 相连的边
7:	**EndWhile**
8:	**Return** C

- 思路：令 A={(u, v)|(u, v)是算法第 4 步选中的边}。由于第 6 步删除了 E'中所有与(u, v)邻接的其他边，则 A 中没有相邻边，又因它覆盖了所有顶点，因此输出结果是图的极大匹配算法。
- 近似比：显然第 5 步每次增加 2 个节点到 C，得到|C|=2|A|。设 C^*是优化解，则 C^*必覆盖图中所有边，也会覆盖 A，得到|C^*|≥|A|。将|C|=2|A|带入，得|C^*|≥|C|/2，即|C|/|C^*|≤2，近似比为 2 得证。

2.证:设 $cost(S^*)$是问题的优化解，则题设问题等价于证明 $cost(S)/cost(S^*)$≤2。下采用反证法，如果 $cost(S)/cost(S^*)>2$，得到 $cost(S^*)<cost(S)/2$，那么围绕 S 中每个点 s，作半径为 $cost(S)/2$ 的圆，必有 1 个 $s^*\in S^*$。假设 p'是离 s^*最近的点，则有 $cost(S)=dist(p',S)\leqslant dist(p',s)\leqslant dist(p',s^*)+dist(s^*,s)$，关系如下图所示，很容易由三角形两边之和大于第三边得到。由于 $cost(S)=dist(p',s)$，又圆心为 s 的圆的半径为 $cost(S)/2$，则 $dist(p',s^*)+dist(s^*,s)\leqslant 2cost(S^*)$。综上得到 $cost(S)\leqslant 2cost(S^*)$，与假设矛盾，命题得证。



3. 算法 Approximate_Multi_Cut(G)

输入 :无向加权图 G=(V, E)，终点集合 S={s_1, s_2, …, s_k}⊆V

输出 :边权最小的多向割集合 C

1:	**For** i←0 to k **DO**
2:	计算 s_i 的最小权孤立割，记为 C_i
3:	**EndFor**
4:	**Return** C=$C_1\cup C_2\cup\cdots\cup C_k$

- 思路：已知第 2 步计算孤立割是一个 P 问题，执行了 k 步，因此该算法也是一个多项式时间算法。删去 C_i 会将 s_i 与其他终点集分开，要将 k 个终点分开，显然删去 k 个终点的孤立割就会得到一个多向割。
- 近似比：设 C^*是 G 的多向割的优化解，令 C^*_i 是分离包含 s_i 的分支与图的剩余部分的割，则 $C^*=C^*_1\cup C^*_2\cup\cdots\cup C^*_k$。因为 C^*中的每条边与两个分支相关联，在计算 k 个 C^*_i 权重和时，恰好是 C^*的 2 倍，即$\sum_{i=1}^{k} w(C^*_i)=2w(C^*)$。显然 C^*_i 也是 s_i 的孤立割，但由于是第 2 步计算的 C_i 是最小权孤立割，则有 $w(C^*_i)\geqslant w(C_i)$。综上，有 $w(C)=\sum_{i=1}^{k} w(C_i)\geqslant$

$\sum_{i=1}^{k}$ w(C_i^*)=2w(C^*)，得到 w(C)/w(C^*)⩾2，近似比为 2。

4.当 $d(c_i, c_p)$⩽εr_p 时，符合题设对圆心和半径的要求。由引理 $d(c_i, c_p) \le r_p/\sqrt{i}$，得 $r_p/\sqrt{i} \le \varepsilon r_p$，进一步 i⩾$1/\varepsilon^2$，因此最小终止值 K 为$\lceil 1/\varepsilon^2 \rceil$。第 4 步排序 P 中点的距离的复杂度为 O($n$log n)，外层循环是关于 $1/\varepsilon^2$ 的多项式，因此算法总的时间是关于 $n\log n 1/\varepsilon^2$ 的多项式，它是一个完全多项式近似模式。

5.将 V 划分为 R 和 L 两部分，组成二分图，即 V=R∪L。假设一共有 n 个顶点，起初设置|R|=n，|L|=0。下面遍历 i 从 2 到 n，做如下 n-1 次操作：令 $k=\lfloor n/i \rfloor$，选择 R 的顶点数为 k 的子集 S，将它加入 L 中，共有 k 个顶点，每个点的度都是 i。这样，R 最少可以只剩 1 个顶点。当 $i=n$ 时，选择 R 的子集，它的度要小于 R 的其余部分。并且对选择的这个集合的每个顶点，它唯一邻接的是我们选择的 R 中的顶点。经过题设过程，选择的顶点集就是 L 中的全部节点。|L|至多为$\sum_2^n \lfloor n/i \rfloor$，求和的结果为 n(log(n+1)-1)。当 n>8 时，有 2|R|=2n⩽n(log (8+1)-1)⩽$\sum_2^n \lfloor n/i \rfloor$⩽|L|。因此，选择 L 作为顶点集，它的顶点数至少是|R|的 2 倍，题设得证。

6.利用贪心策略得到的匹配 M 构造了如下线性规划的可行解 x：

$$\text{Min } \sum_{x_i \in V} X_i, \text{ s.t. } x_i + x_j \geqslant w_{ij},\ x_i \geqslant 0$$

该线性规划的目标函数的最优取值设为 z_{LP}。由于优化解中的边将 2 个顶点分为了 1 组，$z^* = \sum_{(i,j)\in E} w_{ij}$，且每个顶点仅出现 1 次。$z_{LP} = \sum_{x_i \in V} X_i \geqslant \sum_{(i,j)\in E} w_{ij} = z^*$，因此 z_{LP} 是 z^*的上界。得到此可行解的代价至多为 2z，因此 2z⩾z^*，得 z^*/z⩽2，命题得证。