任选20题

Problem 6.1

给定图$G = (V, E, W)$，其中：

$$V = \{v_1, v_2, \cdots, v_n\}, \quad E = \{v_1 v_i | i = 2, \cdots, n\}, \quad W(v_1 v_i) = 1, \quad 2 \leq i \leq n$$

如果将图G作为Prim最小生成树算法的输入，并以v_1为起始点，请问总共要多少次边的权重的比较，来找到与点v_1相关联的最小权值的候选边(candidate edge)？

Problem 6.2

1. 如果Prim算法的输入为一个有n个顶点的完全图，那么Prim算法总共会进行多少次的边权重的比较？

2. 假设顶点为$v_1, \cdots, v_n$，对于任意的i、j $(1 \leq i < j \leq n)$，边权值$W(v_i v_j) = n + 1 - i$。在算法的执行过程中，优先队列中最多可能存在多少个节点？

Problem 6.3

假设用堆来实现优先队列，下面我们对于特定类别的图来分析Prim算法，假设$|V| = n$，$|E| = m$：

1. 度有界图(bounded degree graph)是指一个图中所有的顶点的度最大是k(k是常数)。请对度有界图分析Prim算法；

2. 平面图(planar graph)是指可以以某种方式绘制在一个平面上而没有任何的交叉边的一种连通图。对于平面图，有欧拉公式$|V| - |E| + |F| = 2$，其中$|F|$是当图画在平面上时的面的数量(被边包围的区域，以及一个从最外层边到无限远的开放区域)。例如对于一个简单的三角形，那么它有两个面：一个在三角形里面，一个在外面。外面的面会从各个方向延展到无限远。对于平面图分析Prim算法的时间复杂度。

Problem 6.4

请比较Prim算法和Kruskal算法的好坏。为此你需要对优先队列、并查集的不同实现和图的稠密程度作详细讨论。(注：图的稠密程度可以简单考虑两种情况。如果$m = O(n)$，我们说图是稀疏的；如果$m = \Theta(n^2)$，我们说图是稠密的。)

Problem 6.5

1. 请给出算法用来得到给定的带权重图的最大权重生成树。

2. 我们定义图G的反馈边集(feedback edge set) F：F是图中边集的一个子集，并且G的任意环中至少有一条边来自F。换言之，从G中删除F中的所有边，则G成为一个无环图。请设计一个算法得到给定带权图的最小的反馈边集(这里最小反馈边集的含义是集合中所有边的权重和最小)。

Problem 6.6

给定一个带权图$G = (V, E)$，所有边的权重都为正数，请问对于任一顶点s，下面的情况是否可能：s的一条最短路径树与G的某个最小生成树不共用任何一条边。如果可能，请给出一个例子，否则请证明你的结论。

Problem 6.7
给定一个具有正边权重的图$G = (V, E)$，以及与之对应的一个最小生成树$T = (V, E')$，假定G和T都用邻接表给出。此时若将某条边$e \in E$的权重由$w(e)$修改为$\hat{w}(e)$。在不重新计算整个最小生成树的前提下，通过更新T得到新的最小生成树。请针对以下四种情况，分别给出线性时间的更新算法：

(a) $e \notin E'$且$\hat{w}(e) > w(e)$。

(b) $e \notin E'$且$\hat{w}(e) < w(e)$。

(c) $e \in E'$且$\hat{w}(e) < w(e)$。

(d) $e \in E'$且$\hat{w}(e) > w(e)$。

Problem 6.8
有时我们希望得到一些具有特别性质的“轻”的生成树。以下是一个例子：

- 输入：无向图$G = (V, E)$，边权重w_e，顶点子集$U \subset V$；
- 输出：使得U中节点为叶节点的最轻的(权重最小的)生成树。

  (注：可能还有其它叶节点不是U中节点；所求的最轻生成树不一定存在；所得到的最轻生成树也不一定是最小生成树。)

请针对该问题设计时间为$O((m + n)\log n)$的算法。

Problem 6.9
给定一个有权连通图，和一个特定的边集S(S中没有环)，请给出从G中生成包含S中的所有边的最小生成树。

Problem 6.10
请判断以下叙述或对或错。对于每种情况，请证明你的结论(如果正确)或给出反例(如果错误)。假定图$G = (V, E)$是无向的。如未作特别说明，则图中两条边的权值可能相同。

1. 若G有超过$n - 1$条边，且有唯一一条最重边，则这条边必不属于G的任意最小生成树；
2. 若G中存在一个环，且其上包含了G的唯一最重边e，则e不属于任何最小生成树；
3. 设e是G中的一条最轻边，则e必属于某个最小生成树；
4. 如果图中的最轻边唯一，则该边必属于每个最小生成树；
5. 若G中存在一个环，且该环中的最轻边e唯一，则e必属于每个最小生成树；
6. 两个节点间的最短路径必定是某个最小生成树的一部分；
7. 当存在负权重的边时，Prim算法仍然有效；

Problem 6.11
给定无向连通图$G = (V, E)$，每条边的权重均为正数且各不相同。构建一个图G'，其节点和边与G相同，所不同的是，G'中每条边的权重是图G中对应边的权重的平方。请判断下面的命题是否正确并证明你的结论：T是G的最小生成树当且仅当T是G' 的最小生成树。

Problem 6.12
本题主要考察最小生成树的唯一性。

1. 设$G=(V,E)$是一个无向图。证明如果G所有边的权重都各不相同，则其最小生成树唯一；

事实上，一个更弱的边的权重条件隐含了最小生成树的独特性。

2. 请给出一个带权重的图，它的最小生成树是唯一的，但是它有两条边的权重是相同的；

3. 关于最小生成树的唯一性，我们有如下猜测：最小生成树是唯一的，等价于以下两个条件均成立：

   (a) 将图G的顶点划分为任意两个集合，端点分别位于两个集合中的最小权重边是唯一的；

   (b) 图G的任何圈中的最大权重边是唯一的；

   请证明该猜测的正确性，或者举一个反例否定它；

4. 请给出算法用来判定图是否有一棵唯一的最小生成树。

Problem 6.13
大部分经典的最小生成树算法都是用到安全边(safe edge)和无用边(useless edge)，但是也可以使用其它形式。令G 是一个带权重的无向图，每条边的权重都不相同。如果边e是G中某个圈中的权重最大的边，我们称之为危险(dangerous)的；如果边e不属于G中的任意圈，则称其为有用的(useful)。

1. 请证明有用边在G的最小生成树中；

2. 请证明G的最小生成树中一定没有危险边；

3. “anti-Kruskal”算法是按照权重递减的顺序检测G中的每条边，如果边是危险的，则将其从G中删除。请给出“anti-Kruska”算法的实现，分析算法的复杂度，并考虑是否可以改进算法以降低算法的时间复杂度。

Problem 6.14
现在有一种新的分治算法来计算最小生成树：给定一个图$G=(V,E)$，将顶点集合V划分为两个集合V_1和V_2，使得$|V_1|$和$|V_2|$至多差1，设E_1为一个边集，其中的边都与V_1中的顶点向关联，E_2为另一个边集，其中的边都与V_2中的顶点关联。在两个子图$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$上，分别递归地解决最小生成树的问题。最后，从E中选择一条通过割(V_1,V_2)的最小权边，并利用该边，将所得的两棵最小生成树合并为一棵完整的生成树。请分析该算法能否正确计算出G的最小生成树，并证明你的结论。

Problem 6.15
本题我们讨论图中次小的生成树。

1. 图G的第二小的生成树是指在G的所有生成树中，除了最小生成树以外它的权重和最小。请设计一个算法来计算图G的第二小的生成树；

2. (可选)请设计一个算法来计算带权重的无向图G的第k小生成树。

Problem 6.16
当图中的边权可以为负值时，Dijkstra算法是可能出错的。请构造一个反例说明这一情况。

Problem 6.17
Dijkstra算法所得到的最短路径树是否必然是一棵最小生成树，请证明你的结论。

Problem 6.18
最短路径常常不是唯一的。请设计一个时间复杂度为$O((n+m)\log n)$ 的算法，解决以下问题：

- 输入：无向图$G=(V,E)$，其边权值为正；起始顶点$s \in V$；
- 输出：一个布尔变量数组$U[1..n]$：对于每个顶点v，数组元素$U[v]$取值为TRUE当且仅当从s到u存在唯一的最短路径(注意：$U[s]=TRUE$)。

Problem 6.19
考虑无向图$G=(V,E)$，其边的权重非负。假设我们已经得到了G的一个最小生成树，以及由某个节点$s \in V$到其他所有节点的最短路径。现将所有边的权重加1：

- 最小生成树会发生变化吗？如果变化了，给出一个例子，否则请证明其不变；
- 最短路径会发生变化吗？如果变化了，给出一个例子，否则请证明其不变。

Problem 6.20 (推广的最短路径问题)
在Internet路由问题中，不仅在链路上存在延迟，在路由器上也存在延迟。这一背景引出了对最短路径问题的一个推广：假定一个图除了它的边具有权重$\{l_e : e \in E\}$之外，其顶点同样具有权重$\{c_v : v \in V\}$。现在定义一条路径的权重为其上所有边的权重加上其上所有顶点(包含路径的端点)的权重。给出一个针对以下问题的高效算法：

- 输入：有向图$G=(V,E)$，边权重$l_e > 0$，顶点权重$c_v > 0$；起始顶点$s \in V$；
- 输出：一个数组$Cost[1..n]$，针对每个顶点u，$Cost[u]$是从s到u的最短路径的权值(注：$Cost[s] = c_s$)。

Problem 6.21
考虑这样的一个有向图，其中所有的负边都是从s发出的边：除此之外的其它边权值都为正。以顶点s作为起始点，Dijkstra算法能否适用于本题的情形？请证明你的结论。

Problem 6.22
竞赛图是指有向图$G=(V,E)$中每个点对u,v间有边(u,v)或者(v,u) (两条边不能同时存在)，如图6.1所示。一条有向哈密顿路径是指路径访问有向图中的每个顶点有且仅有一次。

1. 证明对所有的$n \geq 1$，每个有n个顶点的竞赛图有一条哈密顿路径(提示：对顶点个数进行归纳证明)；
2. 请给出在给定的竞赛图上找出哈密顿路径的时间复杂度为$O(n^2)$的算法。

Problem 6.23
给定一组城市，它们之间以高速公路相连，以无向图$G=(V,E)$的形式表示。每条高速公路$e \in E$连接两个城市，公路的长度记为l_e。你想要从城市s 到城市t，但是你的汽车油箱容量有限，在加满的情况下只能行驶L公里。每个城市都有加油站，但城市之间的高速公路上没有加油站。因此，您选择的路径中的每条边(两个城市间的高速公路)e的长度应该满足$l_e \leq L$。

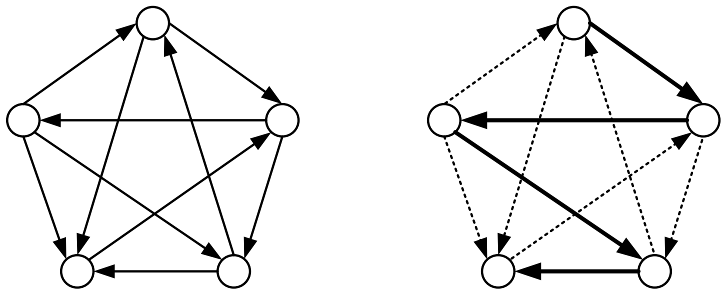

图 6.1: 竞赛图中的哈密顿路径

1. 在给定汽车油箱容量限制的情况下，怎样在线性时间内判断从s到t 之间是否存在一条可行路径；

2. 您现在打算买一辆新车，需要知道从s旅行至t所需的油箱最小容量。给出一个时间复杂度为$O((n+m)\log n)$的算法，计算从s旅行至t所需的油箱最小容量。

Problem 6.24

一个迷宫G中有一个入口和若干出口如图6.2中所标识的“Entry”和“Exit”。在迷宫中每前进一步就是从当前所在的格子走到与其相邻(上、下、左、右四个邻居)的任意一个格子，并且这样的一次前进是具有代价的。现在请考虑下面几个问题：

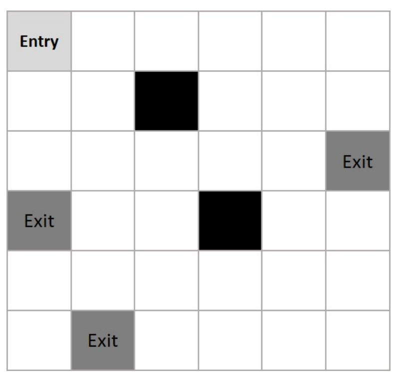

图 6.2: 给定一个入口和若干出口的迷宫

1. 如果在迷宫中每前进一步的代价都是相同的，且都为正数，且迷宫中没有任何障碍，请给出能够找到迷宫中最近出口的算法；

2. 如果在迷宫中每前进一步的代价都是相同的，且都为正数，但是在迷宫中有一些障碍，如图中的黑色格子。请给出能够找到迷宫中最近出口的算法；

3. 如果在迷宫中每前进一步都具有不同的代价，且都为正数。现在有如下两个子问题

   - 假设每次你每次只能向右或者向下前进一步，请给出能够找到迷宫中最近出口的算法；
   - 假设每次你前进的那一步没有上述约束，请给出能够找到迷宫中最近出口的算法；

4. 如果在迷宫中有一些前进的代价是负数，但是不存在负环。假设前进的方向没有约束，请给出能够找到迷宫中最近出口的算法。

Problem 6.25

Floyd-Warshall算法中可以嵌入一些简单的处理来构造一个路由表，这样算法就不仅能计算最短路径的权重，而且能给出具体的路径。我们定义路由表为矩阵GO满足：如果$GO[i][j] = k$，则从v_i到v_j存在一条最短路径，且该路径的第一条边为$v_i \to v_k$。当到达k时，通过查询$GO[k][j]$可以找到路径的下一条边。

1. 请在Floyd-Warshall算法中嵌入构建路由表的逻辑；

2. 上面构造的路由表实际上是一个后继路由表，它总是指明路径上的下一跳应该向何处去。与之对偶地，我们可以构建一个前驱路由表，它总是指明路径的上一跳是从何处而来。请继续在Floyd-Warshall算法中嵌入构建前驱路由表的逻辑。

Problem 6.26

假设有一个输油管道组成的网络，记为图$G = (V, E)$。每根管道uv有容量值$c(u, v)$，它表示这根管道的吞吐率(单位时间可以通过多少油)。对于一个多条管道组成的路径，该路径的吞吐率等于路径上最小管道的吞吐率。而对于给定的起点u和终点v，$cap(u, v)$定义为u到v的所有路径的最大吞吐率。请针对下列问题设计相应的算法：

1. 给定源点s，求它到图中其它所有点的吞吐率；

2. 求图中任意点对之间的吞吐率。

Problem 6.27

给定一个有向图，图中每条边的权重为非负数，和两个不相交的点集S, T，请给出算法找到从S中的任意一个顶点到T中任意顶点的最短路径，算法在最坏情况下时间复杂度为$O(m \log n)$。

Problem 6.28

给定一个强连通有向图$G = (V, E)$，其中每条边的权重都是正数，以及一个特定的顶点$v_0 \in V$。请设计一个高效算法，找出任意一对顶点间的最短路径，并且找出的最短路径必须满足一条额外的限制：这些路径都必须经过顶点v_0。

Problem 6.29

请设计一个算法，以边权值为正的有向图为输入，输出图中的最小权重环的权重值(如果该图是无环的，则输出该图不含环的结论)。要求算法的运行时间最坏情况下为$O(n^3)$。

Problem 6.30

给定一个有向图$G = (V, E)$，其中的边具有权重(权值可以为负)，并且任意两个顶点之间的最短路径最多含有k条边。给出一个算法，在$O(k \cdot m)$时间内找出顶点u和v 之间的最短路径。

Problem 6.31

通常在图中的两个顶点之间存在不止一条最短路径。针对以下任务给出一个线性时间算法：

- 输入：无向图$G = (V, E)$，边的长度为单位长度；顶点$u, v \in V$；

- 输出：从u到v的不同最短路径的数目。