第六章　贪心算法和MST

6.1　练习

Problem 6.1.1
给定一个带权重的图G以及它的最小生成树T。

(a) 请给出当某条边e的权重减小时更新最小生成树的算法。

(b) 请给出当某条边e的权重增加时更新最小生成树的算法。

在上述两个题目中，算法的输入为边e和它的权重；给出的算法需要修改生成树T并且修改后的生成树还是一棵最小生成树。（提示：可以分为$e \in T$和$e \notin T$ 两种情况考虑。）

Problem 6.1.2
假设某个图G中有一棵已经计算出来的最小生成树。如果一个新的顶点及其相关联的边被加入到了G中，该最小生成树可以在多快的时间内被更新呢？

Problem 6.1.3
(a) 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。请给出算法用来得到给定图G的第二小的生成树。

(b) 请给出算法用来得到带权重的无向图G的第k小的生成树。

Problem 6.1.4
(a) 请给出算法用来得到给定的带权重图的最大权重生成树。

(b) 图G的反馈边集（feedback edge set）F是指满足G中任意圈中至少有一条边来自F的图G的边集的子集。换言之，从G中删除F中的所有边，则G成为一个无环图。请给出尽可能快的算法得到给定带权重图的最小的反馈边集。

Problem 6.1.5
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，是否存在某种可能，使得s的一个最短路径树与G的某个最小生成树不共用任何一条边？如果可能，给出一个例子，否则请说明为什么不可能。

Problem 6.1.6
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 6.1.7
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中节点为树叶（可能还有其他树叶节点不是U中节点）的最轻的生成树

（结果不一定是最小生成树。）

给出针对该问题的时间为$O((|E| + |V|) \log |V|)$的算法。（提示：将U中节点从最优解中移除时，会剩下什么？）

Problem 6.1.8
如果从最小生成树中删除掉某一条边就会导致最小生成树的权重增加，则我们称删除掉的边为临界边（critical edge)。请给出在$O(ElogE)$内找到图G中所有临界边的算法。（本题中假设边的权重不需要各不相同）

Problem 6.1.9
在一个带权重的有向图中，存在这样的一条边：删除这条边会使得给定的两个顶点之间的最短路径增加得最多。请给出找到满足要求的边。

Problem 6.1.10
给定一个带权重的连通图，和一个特定的边集S（不含圈），请给出从G中生成包含S中的所有边的最小生成树。

Problem 6.1.11
给定一个连通图G，每条边的权重都不相同。G有n个顶点和m条边。假设e是G中某条边，给出时间复杂度为$O(m + n)$的算法能够确定e是否在G的某个最小生成树中。

Problem 6.1.12
给定图$G = (V, E, W)$，其中$V = \{v_1, v_2, \cdots, v_n\}$，$E = \{v_1 v_i | i = 2, \cdots, n\}$，并且对于$i = 2, \cdots, n$有$W(v_1 v_i) = 1$。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，来找到最小的候选边(candidate edge)？（解决这个问题的过程可能暗示我们，通过保存关于候选边的权重的序的信息可以降低比较的次数。题6.1.13会说明这个过程并不简单。）

Problem 6.1.13

(a) 如果Prim算法的输入为一个有n个顶点的完全无向图，v_1为起始点，那么Prim算法总共会进行多少次的边权重的比较？

(a) 假设顶点为$v_1, \cdots, v_n$，对于$1 \le i < j \le n$，边权值$W(v_i v_j) = n + 1 - i$。在算法执行的某时刻，多少条边是候选边？

Problem 6.1.14
考虑在最小堆中存储候选边（最小堆指堆中每个节点都比它的子节点小）。下面我们对于一般的图以及特定类别的图来评估Prim的MST算法。假设$|V| = n$，$|E| = m$。

1. 对于一般图，基于等式8.1[1]，找出Prim 的MST 算法中要进行的边的权值比较次数在最坏情况下的渐近复杂度。注意当一条候选边被另一条替换的时候，要考虑那些为了decreaseKey所必须的操作。
2. 有界度图（bounded degree graph）是指一个图中所有的顶点的度最大是k（k是常数）。找出Prim的MST算法在一个有界度图中对于边的权值比较次数的关于n的渐近复杂度。
3. "平面图"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面）。例如如果图形是一个简单的三角形，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。对于平面图上的Prim的MST算法，找到对于边的权值比较次数的关于n的渐近复杂度。提示：所有的边都是属于两个面的。

6.2 问题

Problem 6.2.1
以下叙述或对或错。对于每种情况，请证明之（如果正确）或给出反例（如果错误）。其中假定图$G = (V, E)$是无向的。如未说明，不假定边权重各不相同。

1. 若G有超过$|V| - 1$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. Dijkstra算法所得到的最短路径树必定是一个最小生成树。
7. 两个节点间的最短路径必定是某个最小生成树的一部分。
8. 当存在负权重的边时，Prim算法仍然有效。
9. 如果e属于G的某个最小生成树，则其必是跨越G的某个割(cut)的最轻边。（选做）

Problem 6.2.2
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'，每条边的权重是图G中对应边的权重的平方。请判断下面的断言是否正确并简要说明理由：T是G的最小生成树当且仅当T是G' 的最小生成树。

[1]请参考Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析)第395页等式8.1

Problem 6.2.3
设T为图G的一个最小生成树。给定G的一个连通子图H，证明$T \cap H$是H的某个最小生成树的一部分。

Problem 6.2.4
令$G = (V, E)$是任意的一个带权重的连通图，边的权重各不相同。

(a) 证明将V划分为任意的两个子集S, U。端点分别位于两个点集S, U中的权重最小的边一定在G的最小生成树中。

(b) 任何G的圈中的最大权重边都不在G的最小生成树中。

(c) 现在有一个断言是G的最小生成树包含G的每个圈中的最小权重边。请判断断言是否正确，若正确给出证明，否则请给出反例。

Problem 6.2.5
假设图中的边的权重各不相同，那么得到的最小生成树是唯一的。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。

(a) 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。

(b) 请证明当一个带权重的图满足以下两个条件的时候，最小生成树是唯一的。

- 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。
- 图G的任何圈中的最大权重边是唯一的。

(c) 最小生成树唯一的图G中是否必须满足(b)中的两个条件，如果不是，请给出反例。

(d) 请给出算法用来判定图是否有一棵唯一的最小生成树。

Problem 6.2.6
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），但是也可以使用其他形式。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不属于是G中的任意圈，则称其为有用的（useful）。

(a) 请证明有用边在G的最小生成树中。

(b) 请证明G的最小生成树中一定没有危险边。

(c) “anti-Kruskal”MST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”MST算法的实现，并分析算法的复杂度。并考虑是否可以改进算法以降低算法的时间复杂度。

Problem 6.2.7
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。

Problem 6.2.8 (Priority Queue的不同实现)
当priority queue由下列的数据结构实现时，Prim算法和Dijkstra算法的时间复杂度是多少？

- Array[2]
- Binary heap
- （可选）Fibonacci heap[3]

Problem 6.2.9 (Union-Find Set的不同实现)
当并查集由下面的方法实现时，Kruskal算法的复杂度是多少？（请在边权值排序代价之外明确写出使用并查集带来的代价）

- Simple union + simple find
- Weighted union + simple find
- Weighted union + path-compressing find

Problem 6.2.10 (用点覆盖Interval的问题)
给定由数轴上n个区间$[a_1, b_1], \cdots, [a_n, b_n]$组成的集合$S$; 请给出一个复杂度为$O(n \log n)$找出一个点集$P$使得$S$中的每个区间都至少包含$P$中的一个顶点，并且要求$P$的规模最小的算法，并证明算法的正确性和复杂度分析。

Problem 6.2.11
令X是实数轴上的一组区间组成的集合。X的子集$Y(Y \subseteq X)$中的区间如果能够覆盖X中的所有区间，也就是X任意区间上的任意实数值必属于Y中的某一个区间，那么就称Y是X的*tiling path*。*tiling*覆盖的大小就是区间的个数。

请给出算法用尽可能快地找到X的最小*tiling path*。假设用两个数组$X_L\{1...n\}, X_R\{1...n\}$分别表示所有区间的左端点集合和右端点集合。如果你设计的是贪心算法，那么请证明算法的正确性。

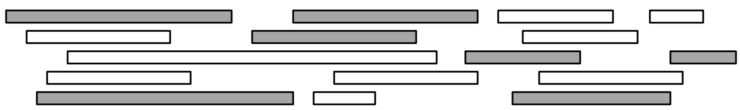


图 6.1: 一组实数上的区间集合，7个阴影区间组成*tiling path*

Problem 6.2.12
令X是实数轴上的一组区间组成的集合。我们称点集P是X的*stabs*，如果X的每个区间至少包含一个P中的顶点。请给出算法能够在尽可能短的时间以内找到X的最小*stabs*。假设用两个数组$X_L\{1...n\}, X_R\{1...n\}$分别表示所有区间的左端点集合和右端点集合。如果你设计的是贪心算法，那么请证明算法的正确性。

Problem 6.2.13
令X是实数轴上的一组区间组成的集合。X的一个适当的着色是指给X中的每个区间着色，任意两个有重叠部分的区间着不同的颜色。请给出高效的算法能够找到X的最小着色数。
假设算法的输入是$L[1...n], R[1...n]$，$L[i], R[i]$分别代表第i个区间的左右两个端点。如果你使用的是贪心算法，请证明算法的正确性。

[2]请参考Sara Baase and Allen Van Gelder. Computer Algorithms-Introduction to Design and Analysis(算法设计与分析)第396页

[3]请参考T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to algorithms(算法导论)第506页

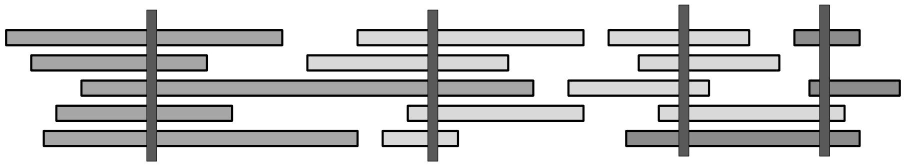


图 6.2: 一组区间的集合

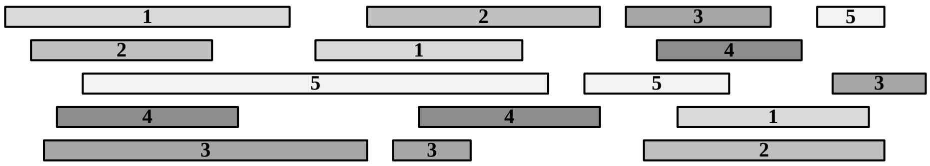


图 6.3: 用5种颜色可以为区间集合着色

Problem 6.2.14

无向图G的一棵瓶颈生成树T是这样的一棵生成树，它最大的边权值在G的所有的生成树中是最小的。瓶颈生成树的值为T中最大权值边的权。

(a) 论证：最小生成树也是瓶颈生成树。

(b)
- 给出一个线性时间的算法，它在给定的图G和一个整数b的情形下，确定瓶颈生成树的值是否最大不超过b。
- 利用你在上一小题中给出的算法作为一个线性时间的，用于求解瓶颈生成树的算法的子例程。

Problem 6.2.15

现在有一种新的分治算法来计算最小生成树。该算法是这样的：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。请论证该算法能正确地计算出G的最小生成树，或者给出一个使该算法不能正确工作的例子。

Problem 6.2.16

沿着一条笔直的公路稀疏地分散着一些房子。用$x_1 < x_2 < ... < x_n$来表示这些房子的位置。现在想要沿着公路设置一些基站使得每个房子在距离其中一个基站不超过t。请设计算法实现这一目标，算法的时间复杂度为$O(n)$，并且证明算法产生的结果是最优的。（证明必须严谨，例如，说明任何和贪心算法得到的方案不同的最优方案都可以转化为和贪心算法所得的解接近的规模相同的解。）

Problem 6.2.17

$Farmsworth$教授从College Park（马里兰大学帕克分校）沿着I-95（95号州际公路）开电动汽车去Miami Florida（佛罗里达州迈阿密市）。充满电的汽车可以行驶100英里，教授出发时汽车是充满电的。沿着公路有一些可以充电的休息站的位置为$x_1 < x_2 < ... < x_n$（按距College Park的距离从小到大排列）。请给出求解到达迈阿密的图中不会出现汽车电量用尽且充电次数最少的算法并证明算法的正确性。

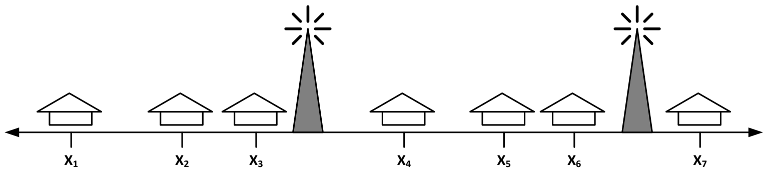
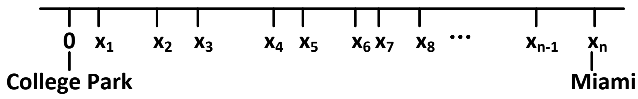


图 6.4: 示意图



Problem 6.2.18

你在一个有n个代表的政治会议会场内，每个代表都隶属于一个政党，但是并不知道他们属于哪个政党。假设你直接询问代表，他会直接拒绝回答，但是你可以通过介绍两个代表认识来分辨他们是否属于一个政党（因为同一政党的代表会礼貌地握手并给予对方微笑；不同政党的代表会怒视对方）。

(a) 假设代表中的大多数（一半以上）来自同一政党（称之为主要政党）。请描述算法用来判定每个代表是否属于这个主要政党。

(b) 假设代表们来自k个政党，一个政党占多数当且仅当属于它的代表的数目比其他任何政党的代表都多。请给出尽可能快的算法找出一个来自占多数的政党的代表。

Problem 6.2.19

Alice想要举办一个舞会，为此需要决定邀请什么人参加。目前共有n个人可供选择，Alice根据他们之间是否相识列出了一个相互配对的列表。她希望邀请尽可能多的人参加，但同时必须考虑以下两点：在舞会上，每个人至少可以各找到5个相识和5个不相识的人。

请就此问题给出一个高效的算法，以n个人的列表及其相识配对列表为输入，输出最优的被邀请客人名单。并基于变量n估算其运行时间。

Problem 6.2.20

一台服务器当前有n个等待服务的顾客。假设我们事先已经掌握了每个顾客所需的服务时间，记顾客i所需的服务时间为t_i分钟。于是若，举例来说，顾客是按照i的数字升序接受服务的，则第i个顾客在得到服务前必须等待$\Sigma_{j=1}^{i} t_j$分钟。

我们希望能够使得总的等待时间

$$T = \Sigma_{i=1}^{n}(\text{第}i\text{个顾客的等待时间})$$

最小。请给出一个有效的算法计算顾客接受服务的最优顺序。

Problem 6.2.21 (研究问题，选做)

请给出时间复杂度为$O(n)$的得到有n个顶点的平面图的最小生成树算法。（提示：收缩平面图中的一条边可以产生另外一个平面图；任何有n个顶点的平面图至多有$3n-6$ 条边。）

第七章　图中的路径

7.1　练习

Problem 7.1.1

设计并分析这样一个算法：以一个无向图$G = (V, E)$为输入，确定G中是否含有一个简单环（即一个不与自身交叉的环），环的长度为4。算法的运行时间最差情况下为$O(|V|^3)$。

你可以随意选择输入图的表示方式：邻接矩阵或是邻接表，只要能使你的算法更简单就好。

Problem 7.1.2

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。

Problem 7.1.3

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$O(k|E|)$时间内找出顶点u和v 之间的最短路径。

Problem 7.1.4

给出一个算法，以边长度为正的有向图为输入，输出图中的最短环的长度（如果该图是无环的，则输出该图不含环的结论）。该算法的运行时间最坏情况下为$O(|V|^3)$。

Problem 7.1.5

给出一个运行时间为$O(|V|^2)$的算法，使其完成以下任务：

输入：一个无向图$G = (V, E)$；边的长度$l_e > 0$；一条边$e \in E$

输出：含有边e的最短环的长度

Problem 7.1.6

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$v_0 \in V$。给出一个高效算法，找出任意一对顶点间的最短路径，还有一个额外的限制：这些路径都必须经过顶点v_0。

Problem 7.1.7

最短路径通常都不唯一：有时对于一个最短路径长度可能会有两条或多条不同的路径。给出一个时间复杂度为$O((|V| + |E|) \log |V|)$ 的算法，解决以下问题：

输入：无向图$G = (V, E)$；边的长度$l_e > 0$；起始顶点$s \in V$

输出：一个布尔变量数组$usp[\cdot]$：对于每个顶点u，数组元素$usp[u]$取值为真当且仅当从s到u存在唯一的最短路径。（注意：$usp[s] = true$）

Problem 7.1.8

考虑无向图$G=(V,E)$，其边权重$w_e \geq 0$。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。

现在将所有边的权重加1，即：新的边权重$w'_e = w_e + 1$。

(a) 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。

(b) 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 7.1.9

Dijkstra的最短路径算法在边的权值为负数的时候是否依然可以正确运作？如果可以，证明你的结论，否则举出反例。

Problem 7.1.10

给定一个加权线图（line graph，无向连通图，除了有两个顶点的度数为1以外所有顶点的度为2）。请给出一个算法能够在线性时间内对图进行处理并且可以在常数时间内返回任意两个顶点之间的最短距离。

Problem 7.1.11

给定一个有向图，如果有一个中间节点v使得存在一条从s到v的路径且路径上的边的权重是严格递增的，存在一条从v到t的路径且路径上的边的权重是严格递减的，则我们称从s到t的路径为$bitonic$（双调）最短路径。请给出算法找到从顶点s到其他顶点的$bitonic$（若存在的话）。（注意这里的路径是简单路径，即路径上没有重复顶点）

Problem 7.1.12

Floyd-Warshall算法可以被修改来构造一个路由表，从其中可以导出最短路径和最短路径的长度。一个矩阵go满足如下条件时是路由表：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,v_k)。当到达k时，通过查询$go[k][j]$可以找到路径的下一跳。

(a) 请你将Floyd-Warshall算法修改来构造一个路由表go。（提示：如果$D[i][j]$因为找到了一条更短的路径而被更新，并且这条更短的路径经过了中间顶点k，那么这条路径的第一跳会是哪个顶点？）

(b) (a)中构造的路由表实际上是一个后继路由表，前驱路由表的定义是对偶的。现在请你继续修改算法，使得算法同时也能够得到前驱路由表。

Problem 7.1.13

有向强连通图$G=(V,E)$的欧拉回路是指通过G中每条边仅一次（但可以访问某个顶点多次）的一个回路。

(a) 证明：图G具有欧拉回路，当且仅当每一个顶点$v \in V$的入度和出度都相等。

(b) 给出一个$O(E)$时间的算法，它能够在图G中存在着欧拉回路的情况下，找出一个回路。

7.2 问题

Problem 7.2.1

给出一种如何在无向图中找出最短环的长度的方法，该无向图中边的长度均为单位长度。

当在深度优先搜索中遇到一条回边（back edge），如(v, w)，则由该回边与其它一些树边（tree edge）能够组成一个从w到v的环。环的长度是$level[v] - level[w] + 1$，其中一个顶点的$level$是指在DFS树中，从根顶点到该顶点的距离。以上思想可以表述成下面的算法：

- 执行一次深度优先搜索，记录每个顶点的$level$值。
- 每当遇到一条回边，计算此时得到的环的长度，如果它比当前最小的环的长度还要小，则更新当前最小环长度为该长度。

通过给出一个反例来证明以上算法并不能保证总是正确，并给出一个简要的说明（一两行文字就可以了）。

Problem 7.2.2

推广的最短路径问题。在Internet路由问题中，不仅在网路上存在延迟，更重要的，在路由器上也存在延迟。这一背景引出了一个推广的最短路径问题。

假定一个图除了它的边具有长度$\{l_e : e \in E\}$之外，其顶点还具有顶点成本$\{c_v : v \in V\}$。现在定义一条路径的成本为其上所有边的长度加上其上所有顶点（包含路径的端点）的成本。给出一个针对以下问题的高效算法：

输入：有向图$G = (V, E)$；边的长度$l_e > 0$，顶点成本$c_v > 0$；起始顶点$s \in V$。

输出：一个数组$cost[\cdot]$，针对每个顶点u，$cost[u]$是从s到u的所有路径的最小成本（即最经济的路径的成本），其中，路径成本的定义如题设中所述。

需要注意的是，$cost[s] = c_s$。

Problem 7.2.3

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$O(ElogV)$。

Problem 7.2.4

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？证明你的结论。

Problem 7.2.5

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7.1所示。一条有向哈密顿路径是指路径访问有向图中的顶点有且仅有一次。

(a) 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）

(b) 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。（你可以用邻接表或者邻接矩阵表示图G，只要时间复杂度满足要求即可）

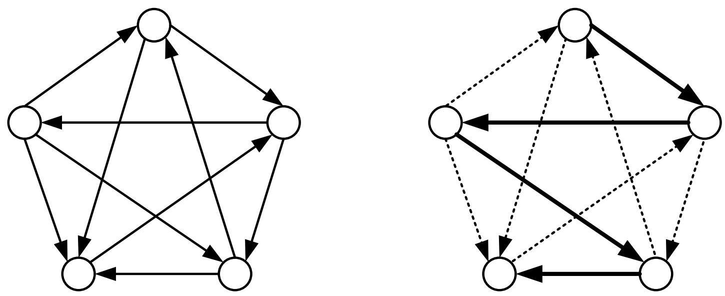


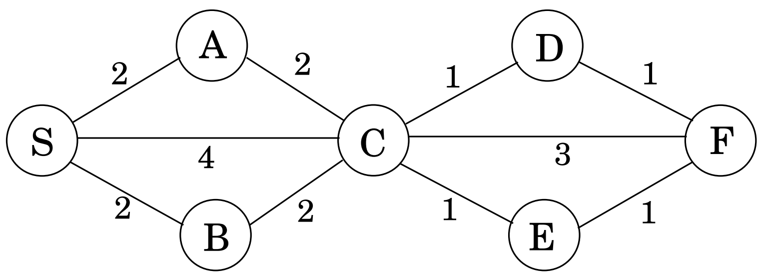
图 7.1: 竞赛图中的哈密顿路径

Problem 7.2.6

当两个顶点间存在多条不同的最短路径（其中，边的长度也不相同），而您需要在其中做出抉择时，最简便的一种方法就是选择一条边数最少的路径。例如，如果用顶点代表城市，边的长度代表在城市间飞行的成本，在同样的成本额度下，从城市s到城市t一定存在很多种飞行方案。从中择一的最简便方式就是选择一种中转最少的。因此，对于一个特定的起始顶点s，定义以下指标：

$$best[u] = \text{从}s\text{到}u\text{的含边最少的最短路径的边的数目。}$$

在下图所示的例子中，顶点S, A, B, C, D, E, F的$best$值分别为$0, 1, 1, 1, 2, 2, 3$。



给出一个针对以下问题的高效算法：

输入：图$G = (V, E)$；边的长度$l_e > 0$；起始顶点$s \in V$。

输出：给出所有顶点$u \in V$的$best[u]$值。

Problem 7.2.7

考虑无向图G中任意两点s, t之间的路径。瓶颈长度（bottleneck length）是指路径上的所有边中的最大权重。瓶颈距离（bottleneck distance）是指从s到t的所有路径的最小瓶颈长度。（如果从s到t没有路径，瓶颈距离就是∞。）

请给出算法用来得到给定无向带权重图中所有点对之间的瓶颈距离，这里所有边的权重都不相同。

Problem 7.2.8

一个有向连通图$G = (V, E)$,每条边有一个权重c（$c(u, u) = \infty$）。给定一条路径$u_1, u_2, ..., u_k$，路径的权重定义为这条路径上的边的权重的最小值。对于图中的任意两个顶点u, v，用$cap(u, v)$来表示从u到v的所有路径的权重的最大值。

(a) 给定一个源点$s \in V$，请给出复杂度为$O(ElogV)$的计算到其他所有顶点的$cap(u, v)$的算法。

(b) 请给出复杂度为$O(V^3)$的计算图中所有顶点对的$cap(u, v)$的算法。

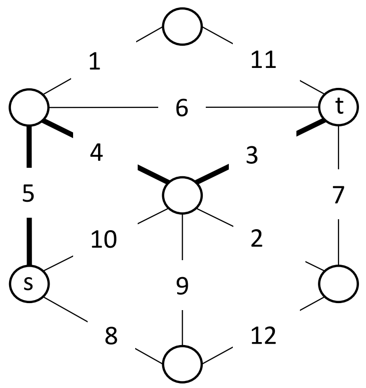


图 7.2: s和t间的瓶颈距离为5

Problem 7.2.9

一组城市（城市构成顶点集合V）之间由一个公路网$G = (V, E)$彼此互相连通。集合E中的每条边对应于一条公路，公路有一个相应的长度l_e。现在需要在该公路网中建设一条新的公路，而新的公路建于两个不同的城市之间。所有可以修建新的公路的两个城市的组合构成一个列表E'。列表E'中可能会修建的每条公路$e' \in E'$具有一个相应的长度。作为公共事业部门的一个设计者，您被要求决定建设哪一条公路$e' \in E'$，使得在公路网G中新建了这条公路之后，两个给定的城市s和t之间的距离得到最大的缩短。给出一个高效算法以解决上述问题。

Problem 7.2.10

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，高速公路的长度以英里记，记为l_e。您想要从城市s 到城市t。存在一个问题：您的汽车油箱容量有限，在加满的情况下只能行驶L英里。每个城市都有加油站，但城市之间的高速公路上并没有加油站。因此，您选择的路径中的每条边（两个城市间的高速公路）的长度应该满足$l_e \leq L$。

(a) 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。

(b) 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$O((|V| + |E|) \log |V|)$的算法，计算从s 旅行至t所需的油箱最小容量。

Problem 7.2.11

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。

(a) 给定源点s，求它到图中其它所有点的吞吐率。

(b) 求图中任意点对之间的吞吐率。

Problem 7.2.12

一个迷宫G里有一个指定入口和一些出口如图7.3中所表示的“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：

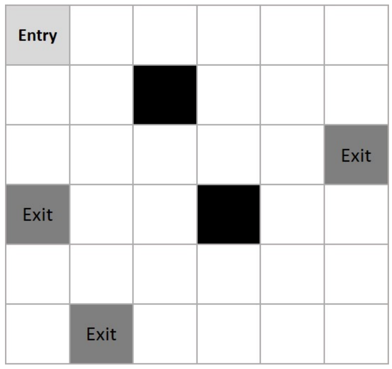


图 7.3: 迷宫

(a) 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。

(b) 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。

(c) 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。

(d) （可选）如果在迷宫中有一些前进的代价是负数，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。