第八周作业-Solution

Lecturer: 杨启哲 Last modified: 2024 年 10 月 31 日

1. F. Lake 教授给出了一种在含负边的有向图中，求顶点 s 到顶点 t 的最短路径的算法。该算法描述如下：

   给图中每条边的权重加一个足够大的常数，使得所有边的权重都为正数。然后用 Dijkstra 算法求解最短路径。最后，将求得的最短路径的长度减去添加的常数，即为原图中 s 到 t 的最短路径长度。

   请问，该算法是否正确？如果正确，请证明；如果不正确，请给出一个反例。

解答. 很遗憾，该算法是不成立的，考虑如下的反例：

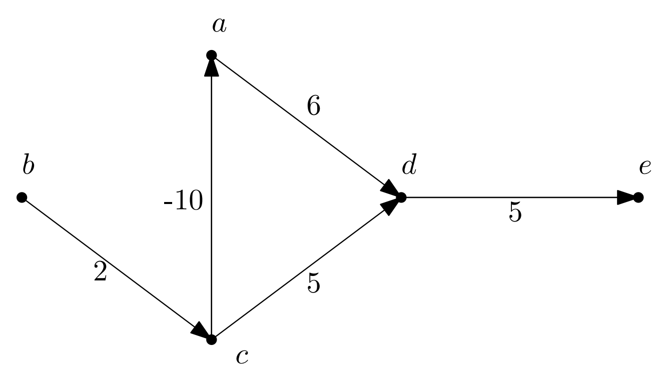

图 1: 最短路径问题的反例

- 在原图中，b 到 e 的最短路径为 $b \to c \to a \to d \to e$，开销为 3.
- 如果每条边的权重加上 11，那么在新的图上 e 到 b 的最短路径为 $b \to c \to d \to e$，开销为 45；实际上的最短路径在新图上的开销则是 47.

□

Remark 0.1

造成这一方法失效的原因在于，最短路径并不是路径边数最短的；但可以看到的是，当这一条件满足时，该方法是成立的；如求最小生成树，因为任何一颗生成树的边的数量都是确定的。

2. 考虑这样一个有向图，其中所有的负边都是从 s 出发的边；除此之外的边的权重都是正数。请问，是否可以用 Dijkstra 算法求解从 s 到其他所有顶点的最短路径？如果可以，请证明；如果不可以，请说明原因。

Remark 0.2

这里假设图中不存在负的环，即最短路径始终都是存在的。

解答. 答案是可以的。原因在于 s 到其他顶点的任何一个最短路径最多只会在第一条边上是负权重的边。我们来严格证明该结论。

设该有向图 $G=(V,E,\omega)$。令 $A=\max_{e=(s,u)\in E}|\omega(e)|+1$，构造新图 $G'=(V',E',\omega')$ 满足:

- $V'=V$，$E'=E$，即顶点集和边集保持不变，
- 对所有由 s 出发的边，定义 $\omega'(e)=\omega(e)+A$，其他边的权重不变。

显然 G' 中的每条边的权重都是正数。下面证明，令 π 是 G' 中 s 到 t 的最短路径，则 π 也是 G 中 s 到 t 的最短路径。

反设结论不成立，即存在一条不同的路径 π' 是 G 中 s 到 t 的最短路径，为了方便描述记 $\omega(\pi')=\sum\omega(e)$ 表示在 ω 下该路径的权重。则由反设条件，$\omega(\pi')<\omega(\pi)$，$\omega'(\pi)\leqslant\omega'(\pi')$。注意到由于 G 中任何一条由 s 出发的路径都恰好有且仅有一条由 s 出发的边，因此我们有:

$$\omega(\pi)=\omega'(\pi)-A,\quad \omega(\pi')=\omega'(\pi')-A$$

从而:

$$\omega(\pi)=\omega'(\pi)-A<\omega'(\pi')-A=\omega(\pi')$$

与反设矛盾。 □

3. 考虑一张没有权重的无向图，通常在这种图中，两个顶点之间不止有一条最短路径。现在请给出一个线性时间算法求出给定的顶点 u 到 v 之间的最短路径的数目。

解答. BFS 可以求出最短路径的长度，我们可以在 BFS 的过程中记录下每个顶点的前驱节点，从而可以得到从 u 到 v 的所有最短路径。具体来说:

(1) 对每个节点我们定义一个数组 $P(u)$，初始 $P(u)=1$，其余都为 0.

(2) 从 u 开始对图进行 BFS，与一般不同的是，每次访问到一条边 (s,t) 时，不管是否 t 访问过，更新 $P(t)$ 为 $P(s)+P(t)$.

(3) 当某一轮访问到 v 的时候，这一轮全部结束后输出 $P(v)$ 的值。

很显然，这个基于 BFS 的算法是线性的。 □

4. 给定一个有权重的有向图 $G=(V,E,\omega)$，其权重可以为负，并且任意两个顶点之间的最短路径最多含有 k 条边，请给出一个 $O(k|E|)$ 的算法，找出从顶点 s 到顶点 t 的最短路径。

解答. 只要理解 Bellman–Ford 算法，这道题就是显而易见的。首先回顾一下 Relax 操作；对一条边 (u, v) 的 Relax 操作是指:

$$\lambda(v) = \min\{\lambda(v), \lambda(v) + \omega((u, v))\}$$

这里 $\lambda(i)$ 表示的是当前到 i 的最短路径。一轮 Relax 操作则是对所有边进行一次上述的更新。因为任何一个最短路径长度至多为 $|V| - 1$，因此在 Bellman–Ford 算法中只要经过 $|V| - 1$ 轮的 Relax 操作，并可得到相应的最短路径 (如果存在)。

在本题的限制下，最短路径长度不超过 k，因此我们只需经历 k 轮的 Relax 操作便可求得相应的最短路径。注意到每一轮 Relax 操作是对所有边都进行一次, 因此该算法的时间复杂性是 $O(k|E|)$.

□

5. 设 $G = (V, E)$ 是一个无向图。图的着色是一个对 G 中顶点的颜色指派问题，使没有两个相邻顶点具有相同的颜色，着色问题是决定对图 G 着色所需要的最少颜色数。考虑下面的贪心方法，其试图解决着色问题。

- 令颜色为 $1, 2, 3, \ldots$
- 尽可能的用颜色 1 对顶点着色。这里尽可能的意思是，使用颜色 1 对**尽可能多的顶点**着色。
- 在剩下的顶点中尽可能的用颜色 2 对顶点着色。
- 再在剩下的顶点中尽可能的用颜色 3 对顶点着色，以此类推，直到所有的顶点都被染色。

请证明，上述贪心算法并不是总能得到最少的着色方案。

解答. 这里尽可能我们理解为每次为尽可能多的顶点上色，我们考虑如下的例子:

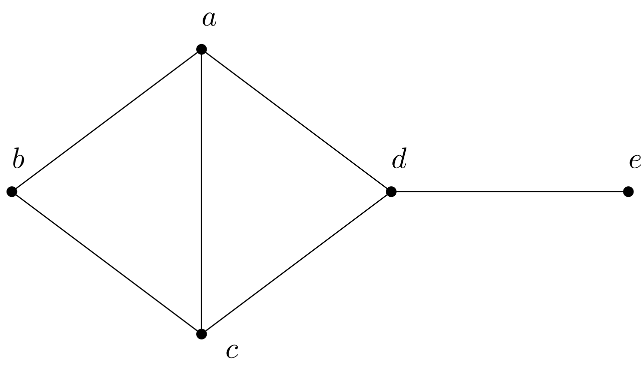

图 2: 着色问题的反例

该例子中，最少所需颜色为 3; 但是按照上述贪心算法，我们可能会得到如下的着色方案:

- b, e 使用颜色 1，剩余每个顶点使用一个不一样的颜色，共需要 4 种颜色。

□