第6章 图与网络分析

6.5 最短路问题

内容安排

- **6.5 最短路问题**
- **6.6 最大流问题**
- **6.7 最小费用流问题**
- **6.8 最大匹配问题**

§6.5 最短路问题

- 实例：给定一个无向图 $G = (V, E)$，边$e \in E$ 上定义有费用（长度）$c_e \geq 0$。有两个特殊的顶点 $s, t \in V$，其中 s 是源顶点，t 是目标顶点。
- 目标：找一条从 s 到 t 的总长度最短的路径。

最短路问题的IP

　　定义指示变量 x_e 为是否使用了边 e。则最短路问题的 IP 可写为：

$$
\begin{aligned}
\min \quad & \sum_{e\in E} c_e x_e \\
\text{s.t.} \quad & \sum_{e\in\delta(S)} x_e \ge 1, \quad \forall S \in \mathcal{S} \\
& x_e \in \{0,1\}, \quad \forall e
\end{aligned}
$$

其中 $\mathcal{S}=\{S \quad V: s \quad S, t \quad S\}$，　(S)是所有一端在 S 中、另一端在 S 外的边的集合，即割$\left(S,\bar{S}\right)$中的所有边。

最短路的LP松弛和它的对偶

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \in \mathcal{S} \\
& x_e \geq 0, \quad \forall e
\end{aligned}
$$

(LP)

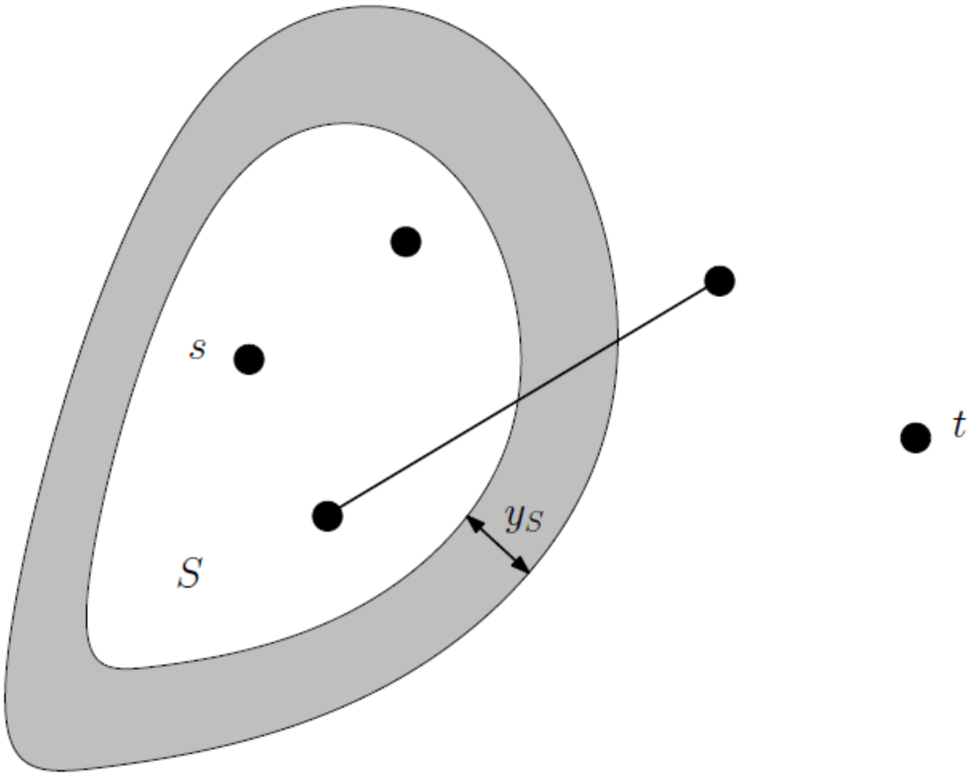
$$
\begin{aligned}
\max \quad & \sum_{S \in \mathcal{S}} y_S \\
\text{s.t.} \quad & \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S \leq c_e, \quad \forall e \\
& y_S \geq 0, \quad \forall S \in \mathcal{S}
\end{aligned}
$$

(DP)

DP的一个几何解释

- 对于S中的每一个S（可称为“城池”），都有一个“护城河”（**moat**）环绕着S。对偶变量y_S即是这个护城河的宽度。
- **DP**的目标函数是挖环绕着每一个S的护城河，使得它们的总宽度最大，以保护每一个S。
- 当然，护城河不能无限地宽。图中的边解释为跨过护城河的“桥”。每条边e之下都有若干护城河S: $e \in \delta(S)$
- 对任一条边e，其跨过的护城河的总宽度不能超过e的长度c_e（参见对偶规划中的约束）。
- 多个连续的桥构成一条路。因此，一条路至少和它所跨过的护城河的总宽度一样长。于是，s、t之间的最短路长度就等于它们之间的护城河的最大宽度。

示例



最短路问题的原始-对偶算法

1. $F = \emptyset$ 。
2. **while** s和t在(V, F)中没有连通时 **do**
3. 　记C为(V, F)中包含s的连通分支。
4. 　增加y_C，直到有一条边 $e \in \delta(C)$成为紧的。
5. 　$F = F \cup \{e\}$。
6. **endwhile**
7. 记P为(V, F)中唯一的s-t路。
8. **return** P。

- 边e是紧的：$\sum_{S:e\in\delta(S)} y_S = c_e$。

引理6.5.1

引理6.5.1：在算法的任一时刻，F都是包含s的一棵树。

- 证明：由于加入F的每条边都是和s连通的，因此F仅包含一个连通分支（即，$C = V(F)$）。
- 当一条边e要加入F时，这条边来自于$\delta(C)$，C是当前F所包含的连通分支。
- 由于e恰好有一个端点在C中，另一个端点在C之外，将e加入F不会形成圈。
- 因此F总是一棵树。□

定理6.5.2

定理 6.5.2：算法找到了一条最短 ***s-t*** 路。

- 证明：因为 **P** 中的每一条边都是紧的，所以有

$$\sum_{e\in P} c_e = \sum_{e\in P}\ \sum_{S\in\mathcal{S}:e\in\delta(S)} y_S = \sum_{S\in\mathcal{S}} |P\cap\delta(S)|\, y_S \text{。}$$

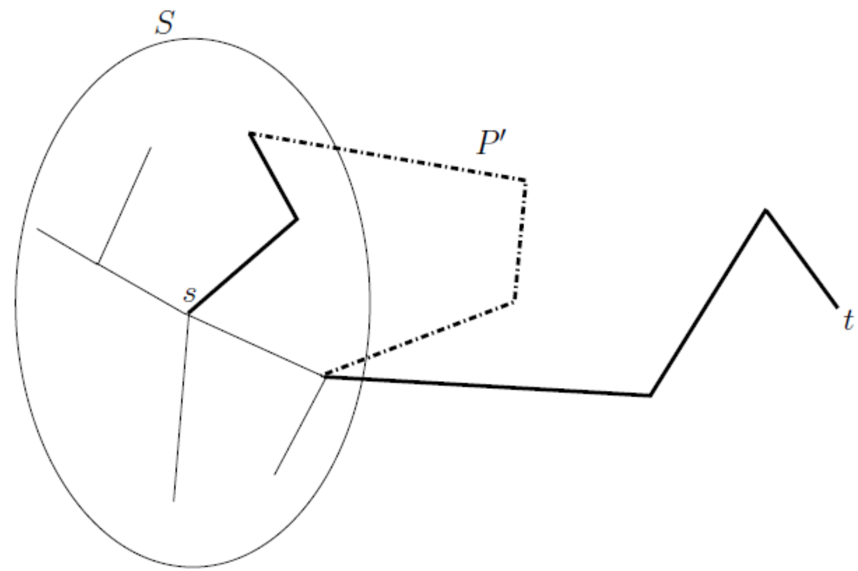
- 断言：只要 $y_S > 0$，就有 $|P \quad (S)| = 1$。
- 先假设这个断言是成立的，稍后给出证明。于是就有

$$\sum_{e\in P} c_e = \sum_{S\in\mathcal{S}} y_S \le \mathrm{OPT} ,$$

从而定理成立，证明就结束了。

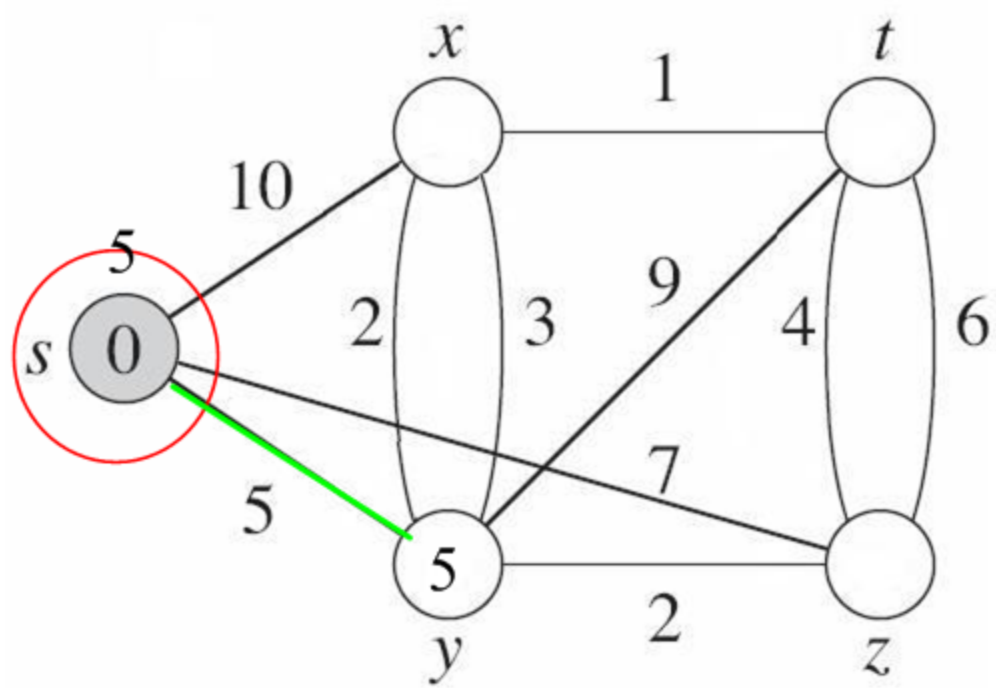
- 反证断言。假设对某个 $S \quad \mathcal{S}$，有 $y_S > 0$ 和 $|P \quad (S)| > 1$。
  这表明 P 穿越了 (S)至少 3 次。

证明



- 因此 P 上就有一条子路径 P'将 S 中的两个顶点连接在了一起。
- 由于 $y_S > 0$，算法必定增加过 y_S。在算法刚要增加 y_S 的时刻，F 是连接（span）S 中所有顶点的一棵树。
- 于是 F　P'中就包含一个圈。这与引理 6.5.1 矛盾。□

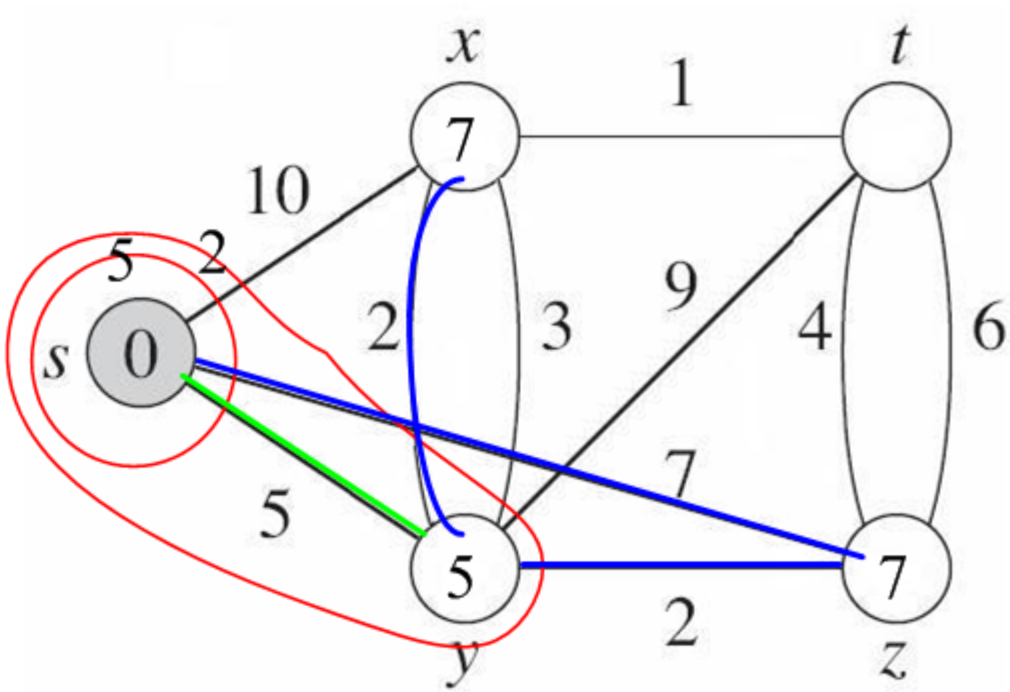
例子



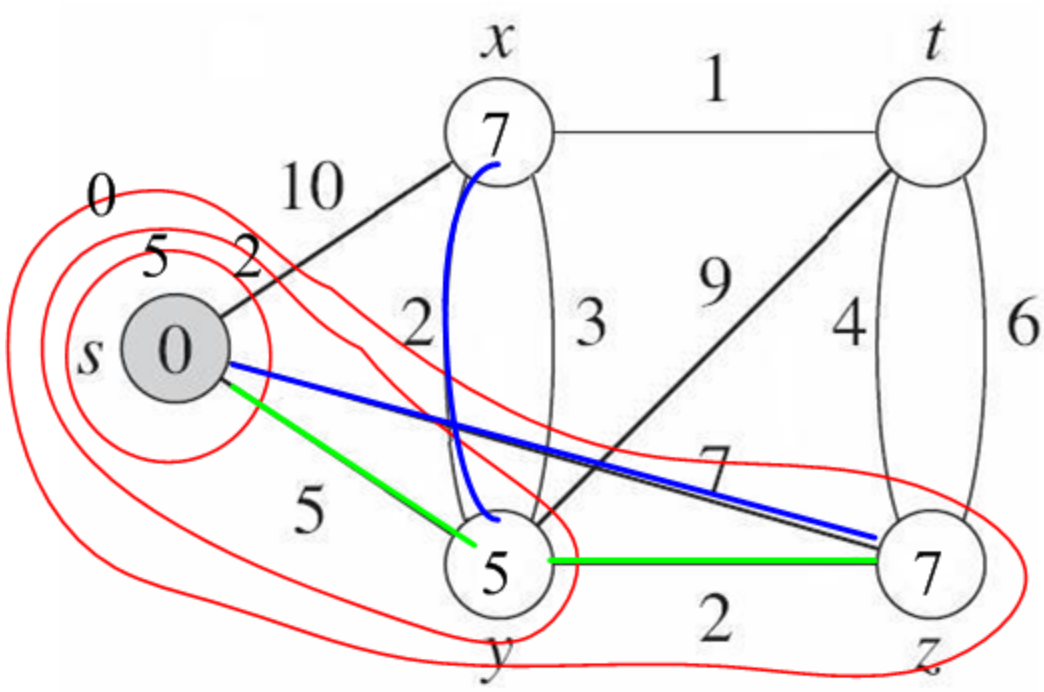
蓝边：紧的边。

绿边：加入到F中的边。

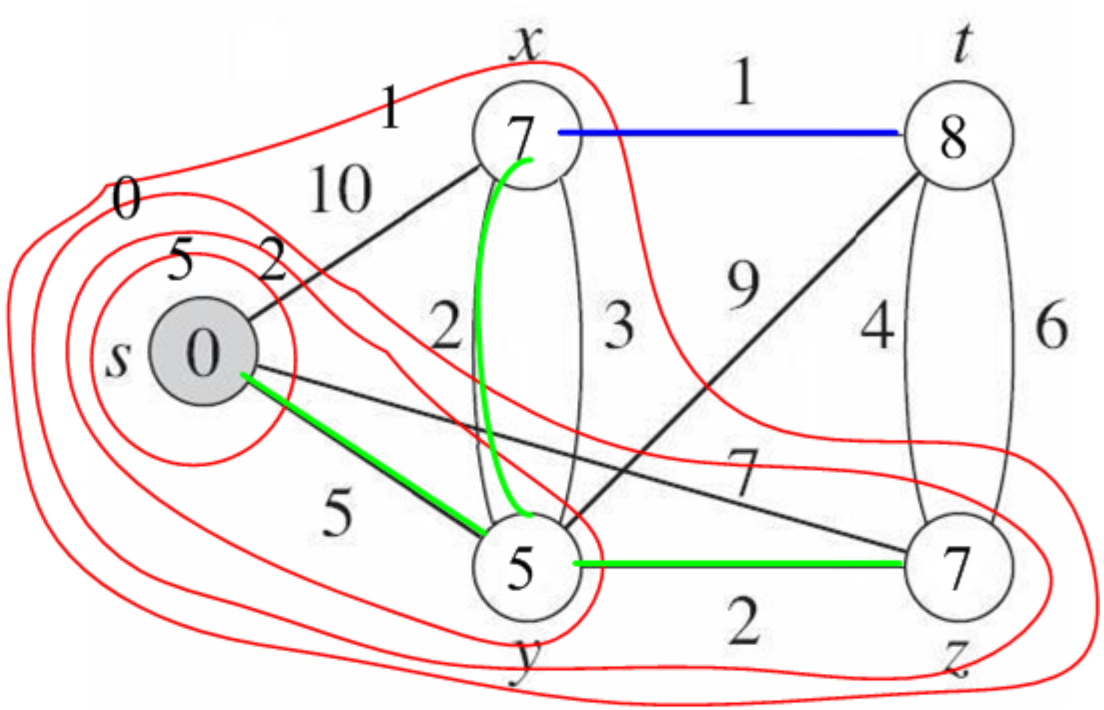
例子



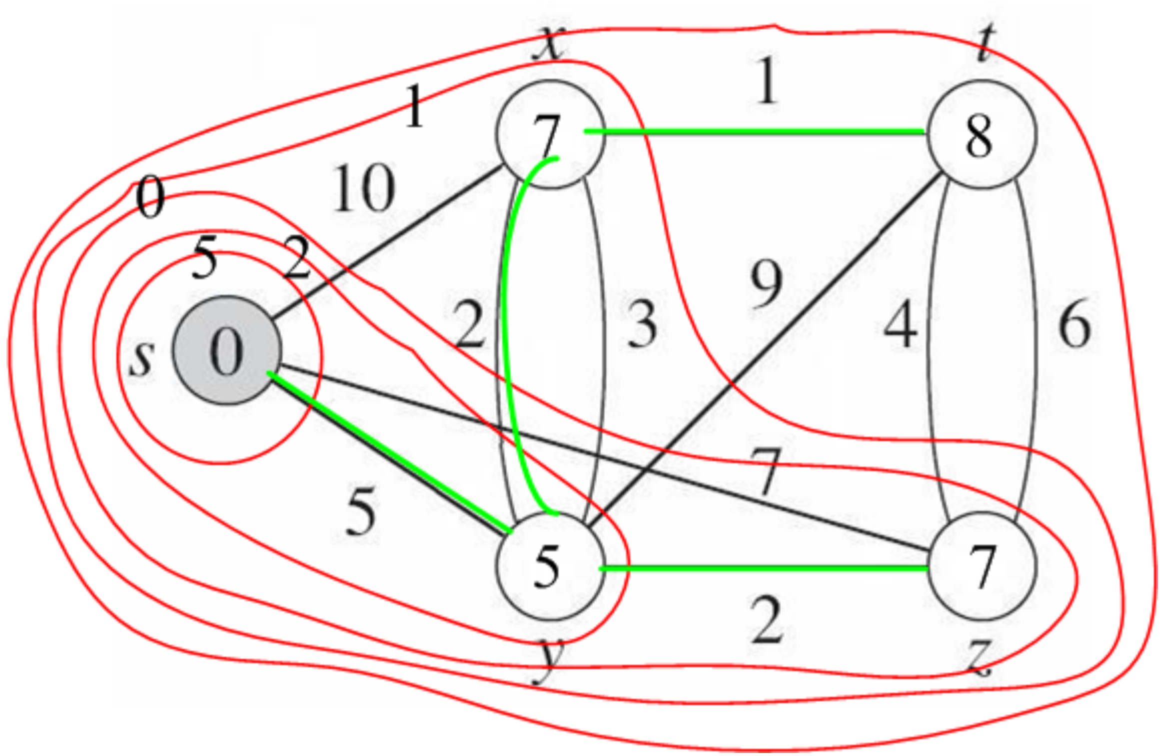
例子



例子



例子

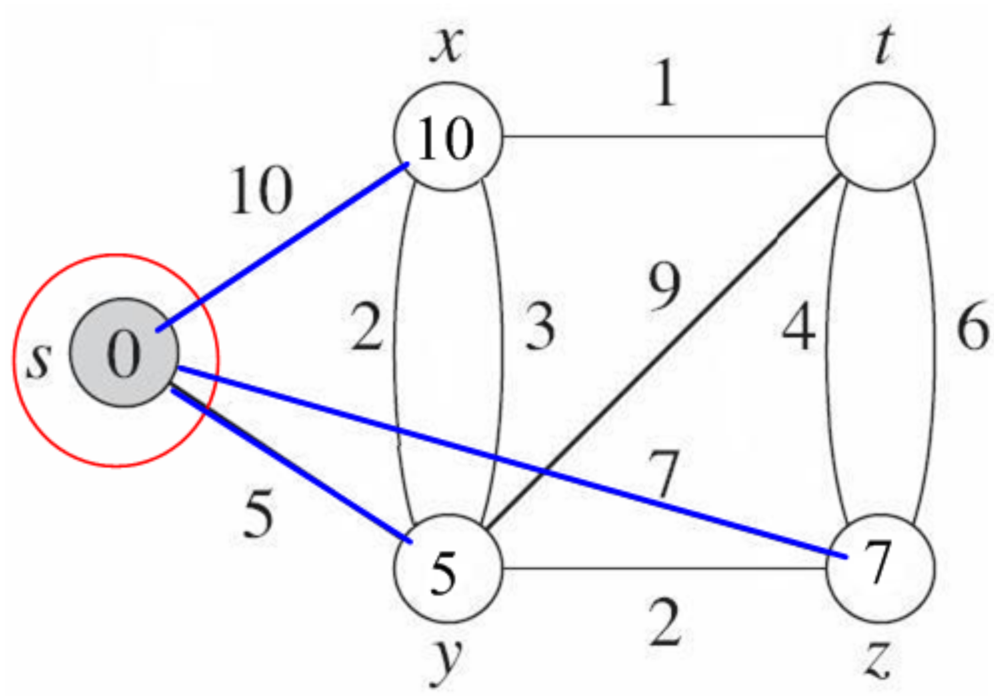


与Dijkstra算法的比较

Dijkstra算法：每次加入距离s最近的一个点。

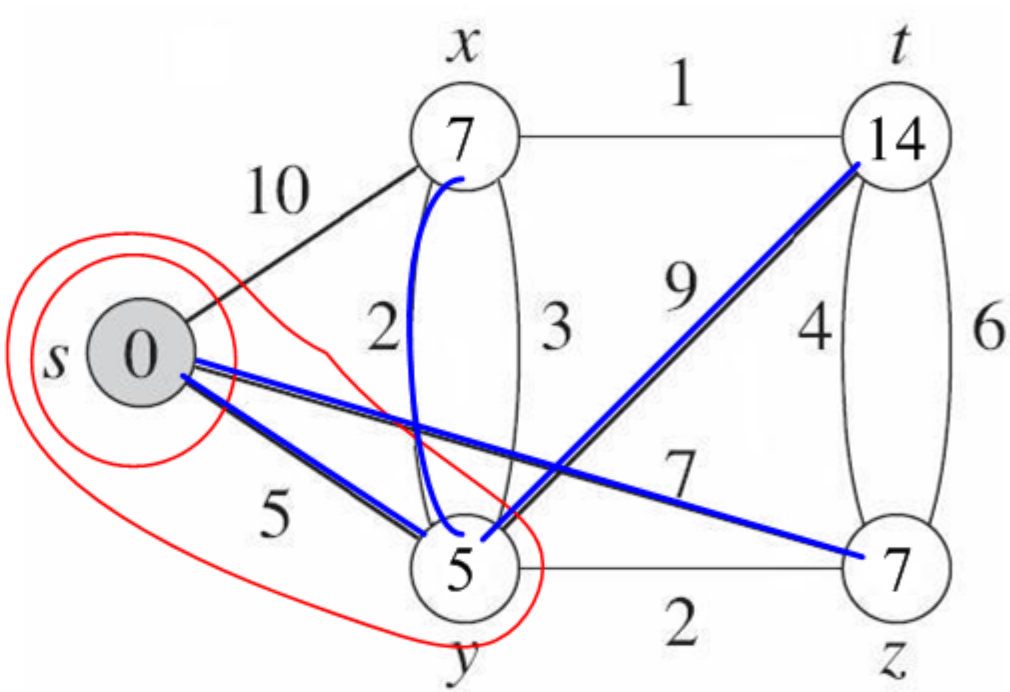
1. **W ={*s*}; *p(s)=0*** 。
2. **for all** $y \in V - \{s\}$ **do** $p(y) = c_{sy}$
3. **while** $W \neq V$ **do**
4. **begin find** $\min\{p(y) : y \notin W\}$**, say *p(x)***
5. **set** $W = W \cup \{x\}$**;**
6. **for all** $y \in V - W$
7. $p(y) = \min\{p(y), p(x) + c_{xy}\}$
8. **end**
9. **end**

与Dijkstra算法的比较

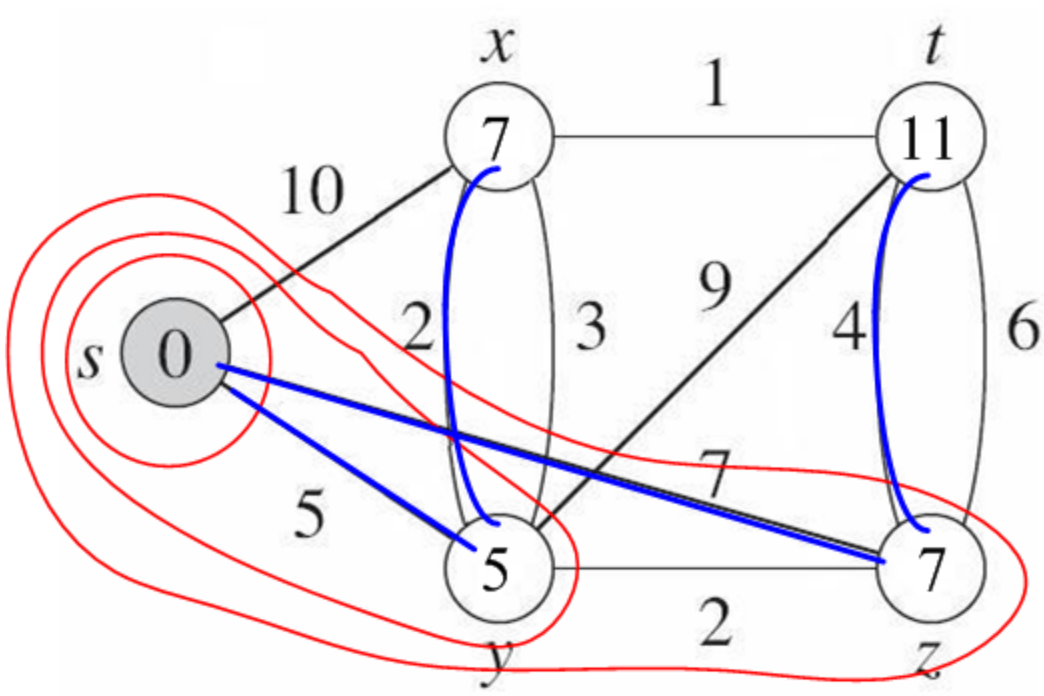


Dijkstra算法：每次加入距离s最近的一个点。

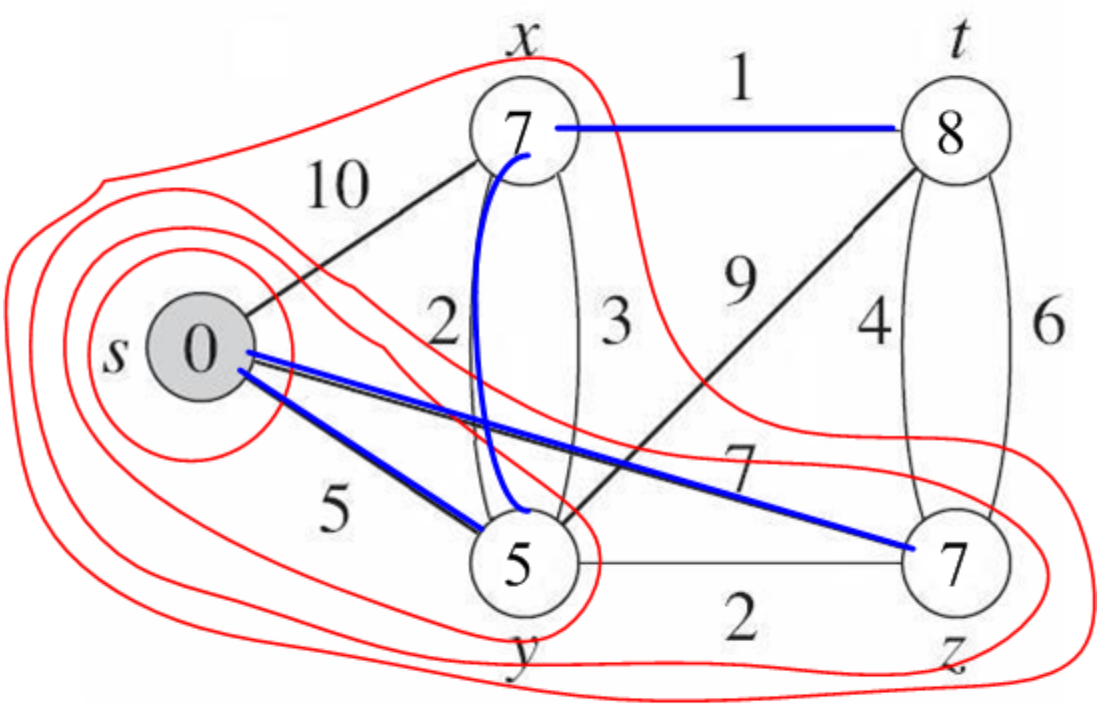
D算法



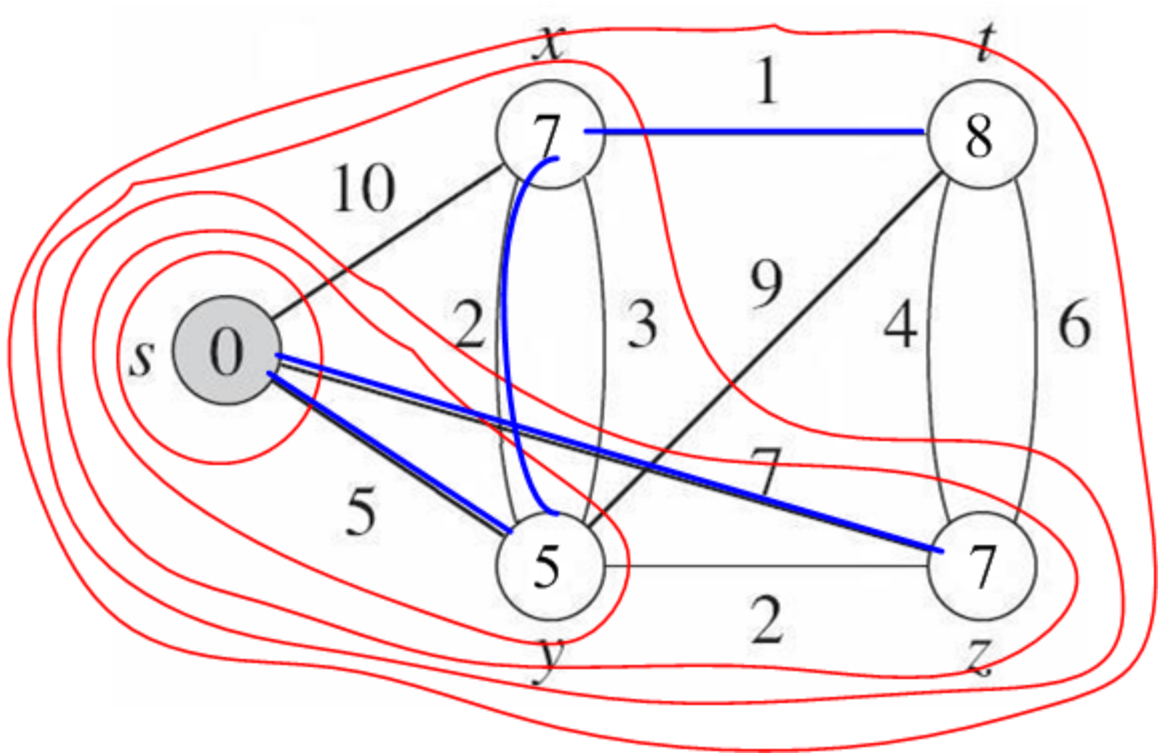
D算法



D算法



D算法



结论：最短路的原始-对偶算法与Dijkstra算法本质上是相同的。

谢谢大家