



贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

贪心算法

讲授者 王爱娟

aijuan321@foxmail.com

重庆理工大学 计算机科学与工程学院

August 22, 2024



目录

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

① 贪心法的思想

② 最小生成树之PRIM算法

③ 最小生成树之Kruskal算法



贪心法的实例

贪心算法

讲授者 王爱娟

目录

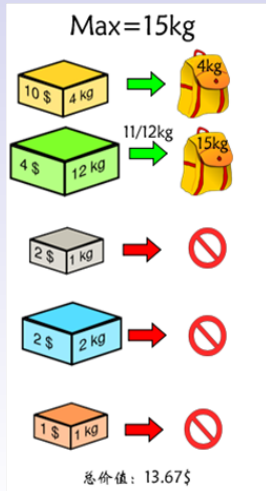
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

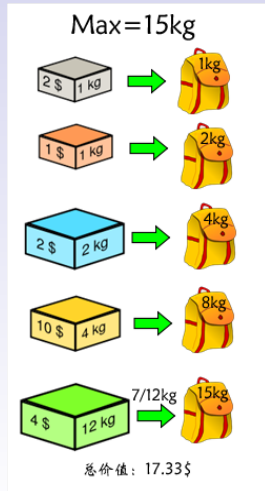
选择1:

按物品价值



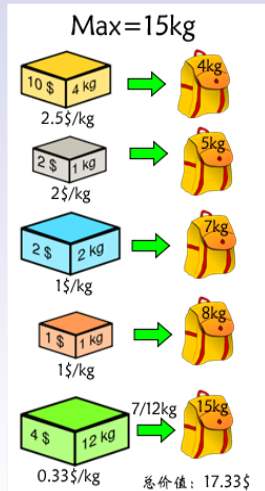
选择2:

按物品重量



选择3:

按物品价值重量比





贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

找零问题：

(1) 硬币面额： $d_1 = 25, d_2 = 10, d_3 = 5, d_4 = 1$



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

找零问题：

(1) 硬币面额： $d_1 = 25, d_2 = 10, d_3 = 5, d_4 = 1$
找零48美分？



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

找零问题：

(1) 硬币面额： $d_1 = 25, d_2 = 10, d_3 = 5, d_4 = 1$

找零48美分？

1个 $d_1 = 25$ ，2个 $d_2 = 10$ ，3个 $d_4 = 1$

选择策略：从当前的几个选择中确定一个最佳选择。

贪心法的思想

在问题的求解过程中

- **可行性：**所做的每一步选择必须满足问题的约束。
- **局部最优：**选择不从全局考虑，仅考虑当前状态(局部状态)。总是做出在**当前看来**是最好的选择
- **不可取消：**选择具备不可取消性，即一旦选择，在算法的后面步骤中就无法改变了！



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$
找零30美分?



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$
找零30美分?
1个 $d_1 = 25$, 5个 $d_3 = 1$



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$
找零30美分?
1个 $d_1 = 25$, 5个 $d_3 = 1$

贪心法的特点

贪心算法不是对所有问题都能得到整体最优解, 但对范围相当广泛的许多问题它能产生整体最优解或者是整体最优解的近似解。



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$
找零30美分?
1个 $d_1 = 25$, 5个 $d_3 = 1$

贪心法的特点

贪心算法不是对所有问题都能得到整体最优解, 但对范围相当广泛的许多问题它能产生整体最优解或者是整体最优解的近似解。

如何证明某一贪婪算法能获得最优解? (本课程不涉及)



贪心法的思想

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

(2) 硬币面额: $d_1 = 25, d_2 = 10, d_3 = 1$
找零30美分?
1个 $d_1 = 25$, 5个 $d_3 = 1$

贪心法的特点

贪心算法不是对所有问题都能得到整体最优解, 但对范围相当广泛的许多问题它能产生整体最优解或者是整体最优解的近似解。

如何证明某一贪婪算法能获得最优解? (本课程不涉及)
其方法:

- 数学归纳法证明算法每一步获得的部分解能够扩展到全局最优解;
- 证明算法在接近目标的过程中, 每一步的选择不会比其他算法差;
- 基于算法的输出, 证明算法得到的解的最优性;



最小生成树

贪心算法

讲授者 王爱娟

目录

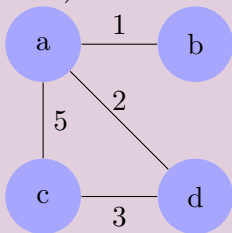
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

生成树

连通图的一棵**生成树**是包含图的**所有点**的连通无环子图(一棵树)



图

最小生成树

加权连通图的一棵**最小生成树**是图的一棵权重最小的生成树



最小生成树

贪心算法

讲授者 王爱娟

目录

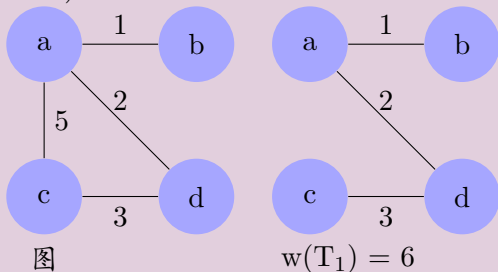
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

生成树

连通图的一棵**生成树**是包含图的**所有点**的连通无环子图(一棵树)



最小生成树

加权连通图的一棵**最小生成树**是图的一棵权重最小的生成树



最小生成树

贪心算法

讲授者 王爱娟

目录

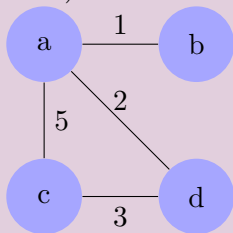
贪心法的思想

最小生成树
之PRIM算法

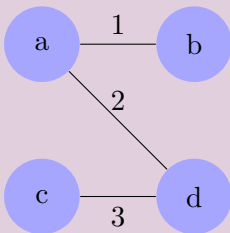
最小生成树
之Kruskal算
法

生成树

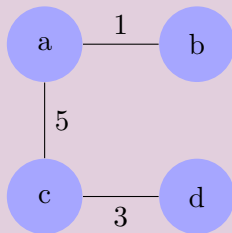
连通图的一棵**生成树**是包含图的**所有点**的连通无环子图(一棵树)



图



$w(T_1) = 6$



$w(T_2) = 9$



最小生成树

贪心算法

讲授者 王爱娟

目录

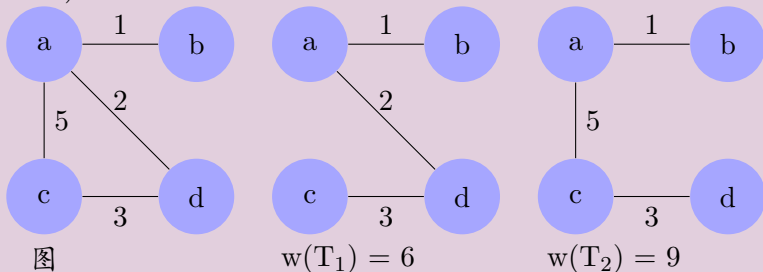
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

生成树

连通图的一棵**生成树**是包含图的**所有点**的连通无环子图(一棵树)



最小生成树

加权连通图的一棵**最小生成树**是图的一棵权重最小的生成树



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。



最小生成树之Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

回顾：图的定义

一个图 G 可以用二元组 $\langle V, E \rangle$ 表示，其 V 表示图的顶点集合， E 表示连接顶点的边集

Prim算法概述

- ① 开始产生最小生成树：从 V 中任意顶点 v_0
 - 顶点集 V 被划分为两个集合： $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树：总是找出权重最小的边 (a, b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 。将 b 加入到 V_T 中，并从 $V - V_T$ 中删除
- ③ 反复执行第2步，直至集合 $V - V_T$ 为空。

为什么终止条件是 $V - V_T$ 为空

最小生成树的顶点与图的顶点相同！

► Jump to definition



Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！



Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

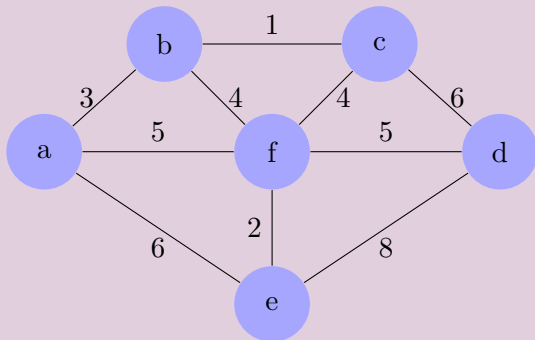
最小生成树
之Kruskal算法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

Prim算法实例

考虑图G如下所示：





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

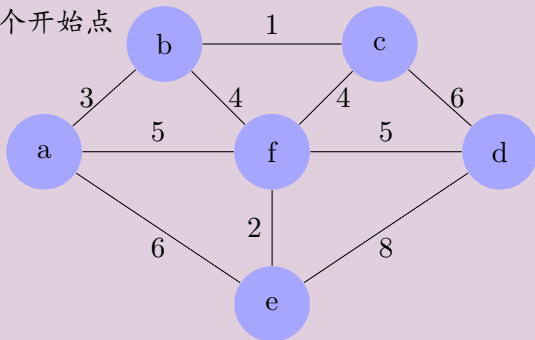
为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

Prim算法实例

考虑图G如下所示：

随机选择一个开始点





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

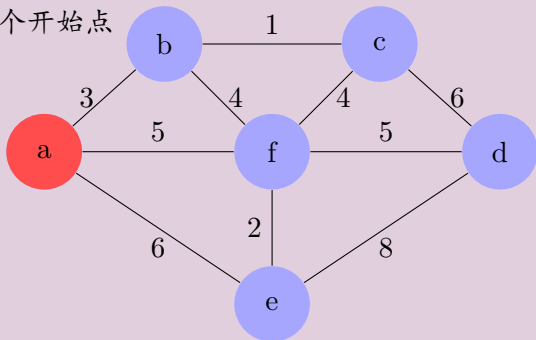
为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

Prim算法实例

考虑图G如下所示：

随机选择一个开始点





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

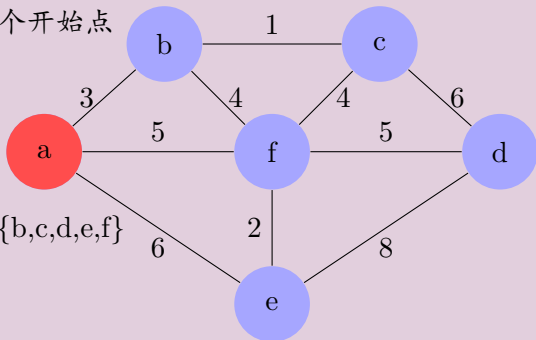
Prim算法实例

考虑图G如下所示：

随机选择一个开始点

$V_T = \{a\}$

$V - V_T = \{b, c, d, e, f\}$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

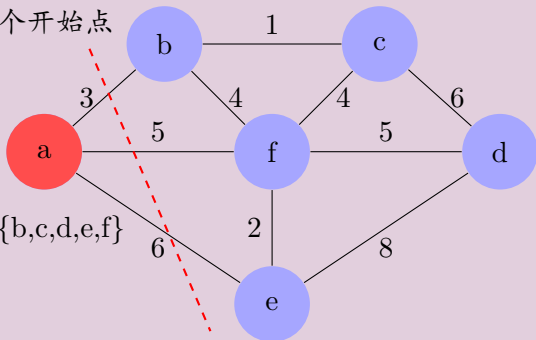
Prim算法实例

考虑图G如下所示：

随机选择一个开始点

$V_T = \{a\}$

$V - V_T = \{b, c, d, e, f\}$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

Prim算法实例

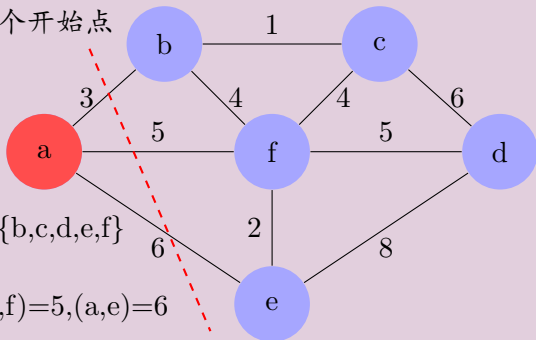
考虑图G如下所示：

随机选择一个开始点

$$V_T = \{a\}$$

$$V - V_T = \{b, c, d, e, f\}$$

$$(a,b)=3, (a,f)=5, (a,e)=6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

为什么称Prim算法为贪心算法

每步总是只选择当前最优的解：最短边！

Prim算法实例

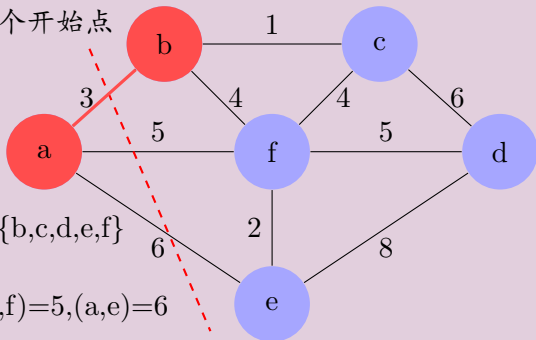
考虑图G如下所示：

随机选择一个开始点

$$V_T = \{a\}$$

$$V - V_T = \{b, c, d, e, f\}$$

$$(a,b)=3, (a,f)=5, (a,e)=6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

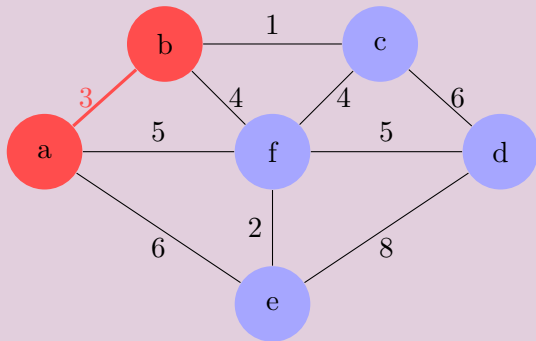
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续1)





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

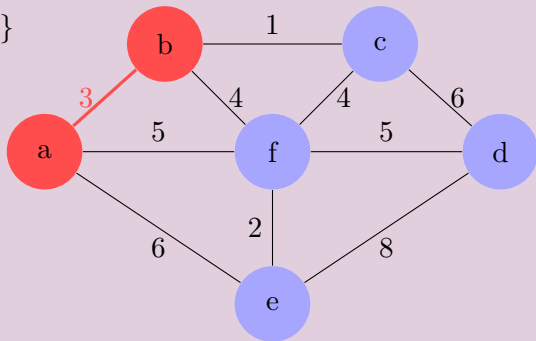
Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续1)

$V - V_T = \{c,d,e,f\}$

$V_T = \{a,b\}$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

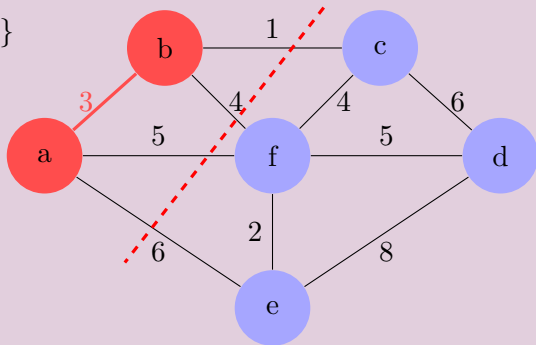
Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续1)

$$V - V_T = \{c, d, e, f\}$$

$$V_T = \{a, b\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续1)

$$V - V_T = \{c, d, e, f\}$$

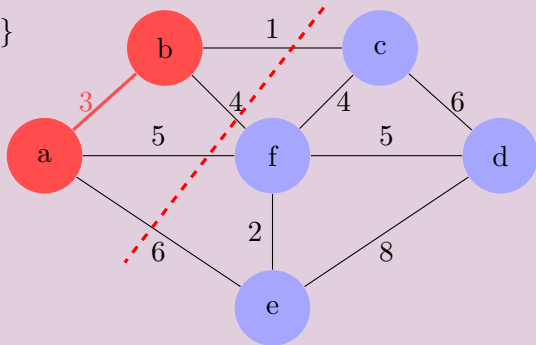
$$V_T = \{a, b\}$$

$$(b, c) = 1$$

$$(b, f) = 4$$

$$(a, f) = 5$$

$$(a, e) = 6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续1)

$$V - V_T = \{c,d,e,f\}$$

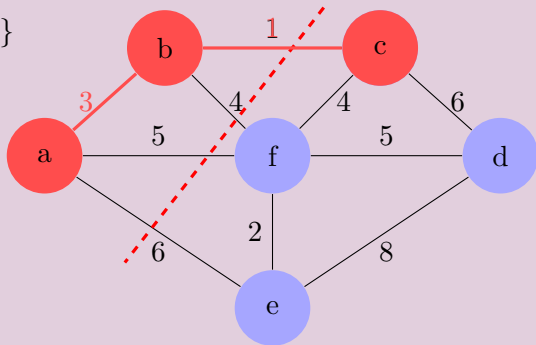
$$V_T = \{a,b\}$$

$$(b,c)=1$$

$$(b,f)=4$$

$$(a,f)=5$$

$$(a,e)=6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

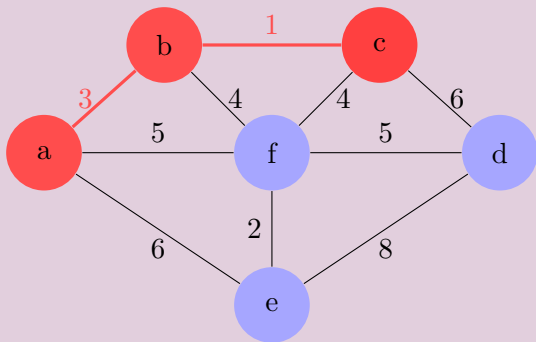
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续2)





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

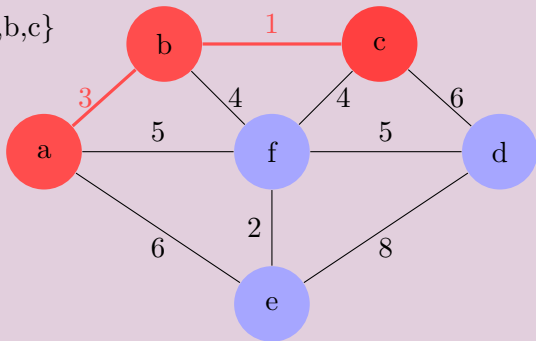
Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续2)

$$V - V_T = \{d, e, f\}$$

$$V_T = \{a, b, c\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

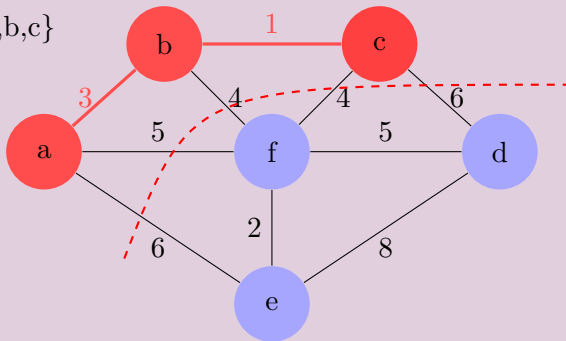
Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续2)

$$V - V_T = \{d, e, f\}$$

$$V_T = \{a, b, c\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续2)

$$V - V_T = \{d, e, f\}$$

$$V_T = \{a, b, c\}$$

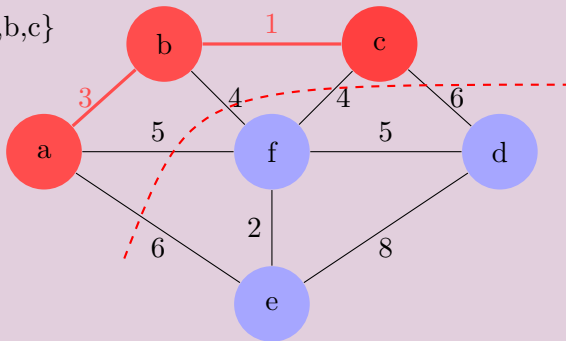
$$(c, f) = 4$$

$$(c, d) = 6$$

$$(b, f) = 4$$

$$(a, f) = 5$$

$$(a, e) = 6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续2)

$$V - V_T = \{d, e, f\}$$

$$V_T = \{a, b, c\}$$

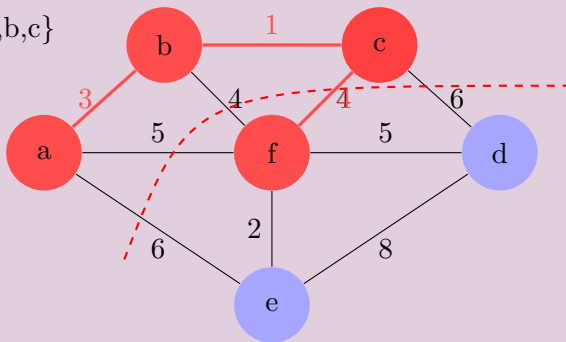
$$(c, f) = 4$$

$$(c, d) = 6$$

$$(b, f) = 4$$

$$(a, f) = 5$$

$$(a, e) = 6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

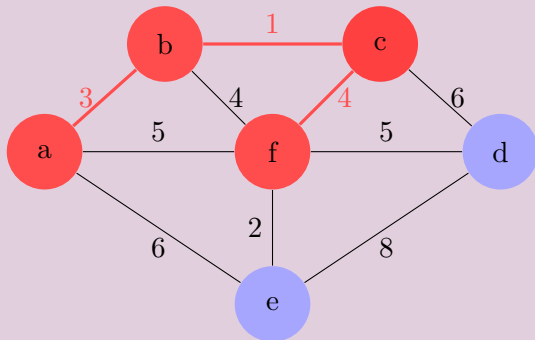
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续3)





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

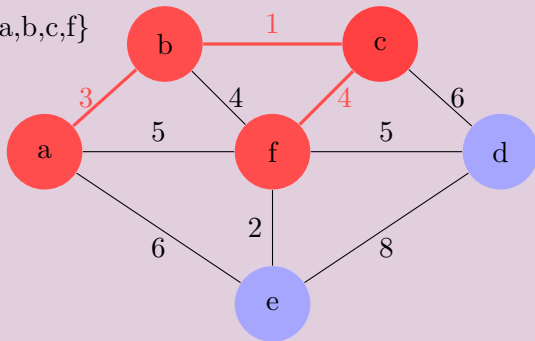
Prim算法核心

总是找出权重最小的边 (a,b) ，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续3)

$$V - V_T = \{d,e\}$$

$$V_T = \{a,b,c,f\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

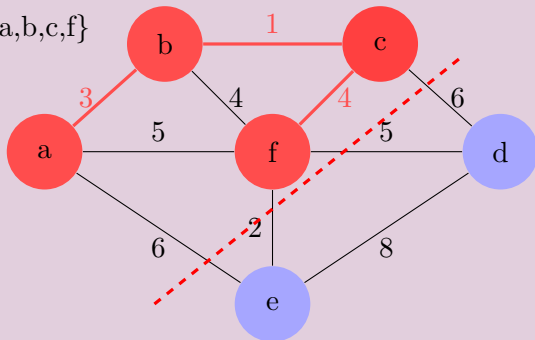
Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续3)

$$V - V_T = \{d, e\}$$

$$V_T = \{a, b, c, f\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续3)

$$V - V_T = \{d, e\}$$

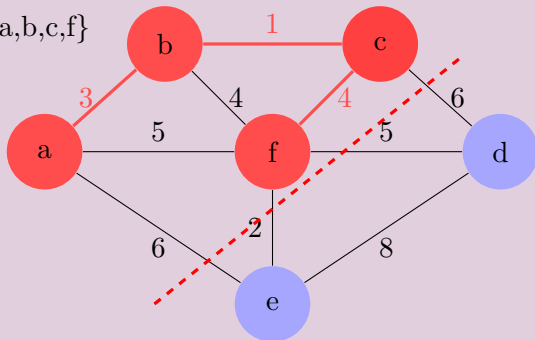
$$V_T = \{a, b, c, f\}$$

$$(c, d) = 6$$

$$(f, d) = 5$$

$$(f, e) = 2$$

$$(a, e) = 6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b)，其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续3)

$$V - V_T = \{d, e\}$$

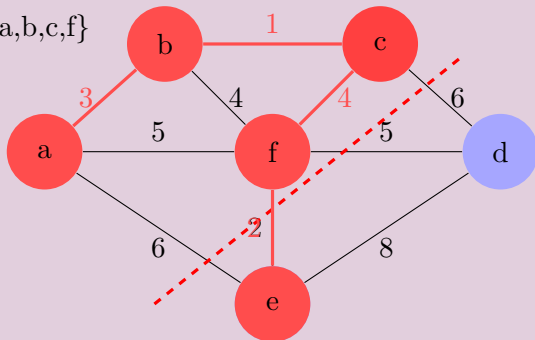
$$V_T = \{a, b, c, f\}$$

$$(c, d) = 6$$

$$(f, d) = 5$$

$$(f, e) = 2$$

$$(a, e) = 6$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

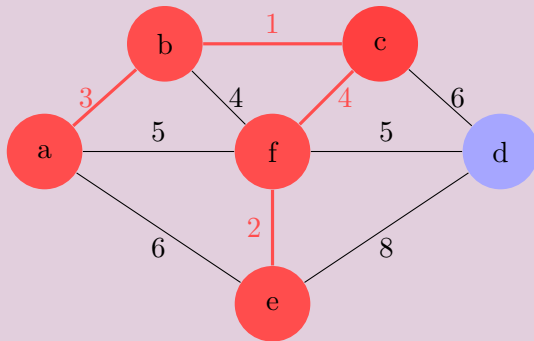
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续4)





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

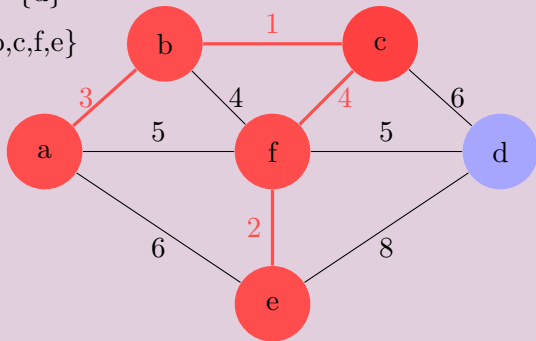
Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续4)

$$V - V_T = \{d\}$$

$$V_T = \{a,b,c,f,e\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

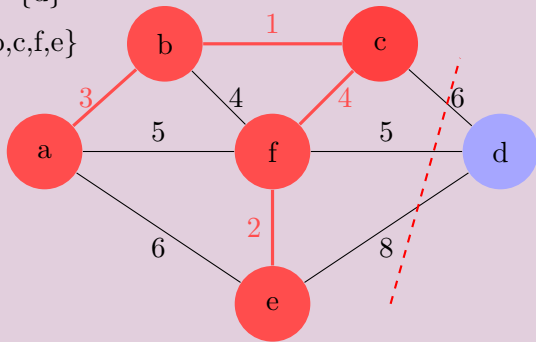
Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续4)

$$V - V_T = \{d\}$$

$$V_T = \{a,b,c,f,e\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续4)

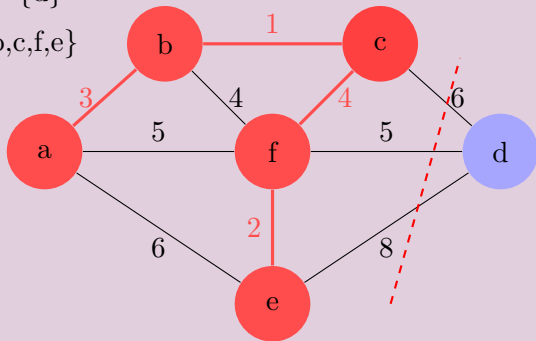
$$V - V_T = \{d\}$$

$$V_T = \{a,b,c,f,e\}$$

$$(c,d)=6$$

$$(f,d)=5$$

$$(e,d)=8$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续4)

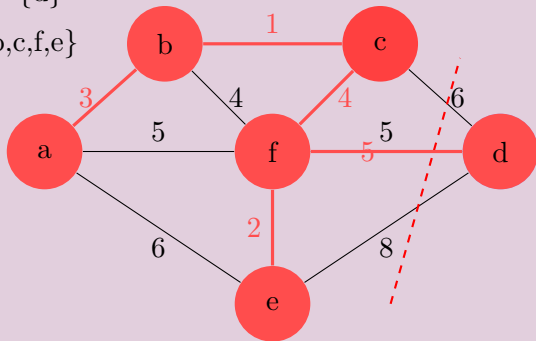
$$V - V_T = \{d\}$$

$$V_T = \{a,b,c,f,e\}$$

$$(c,d)=6$$

$$(f,d)=5$$

$$(e,d)=8$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

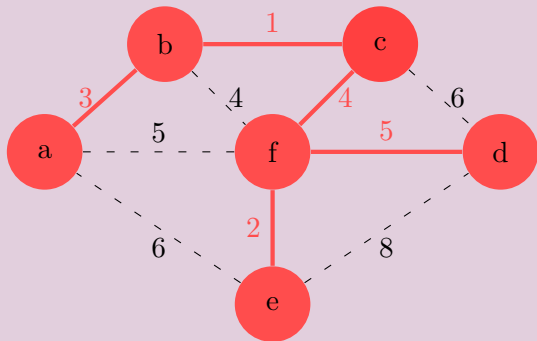
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续5)





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

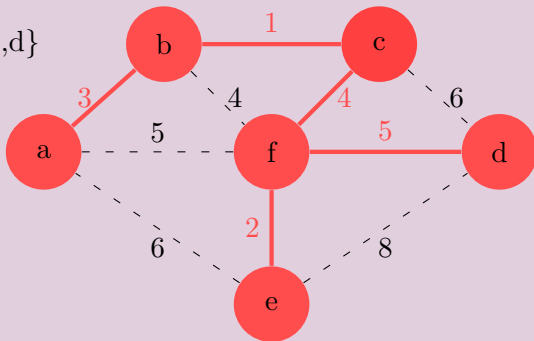
Prim算法核心

总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续5)

$$V - V_T = \emptyset$$

$$V_T = \{a, b, c, f, e, d\}$$





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

Prim算法核心

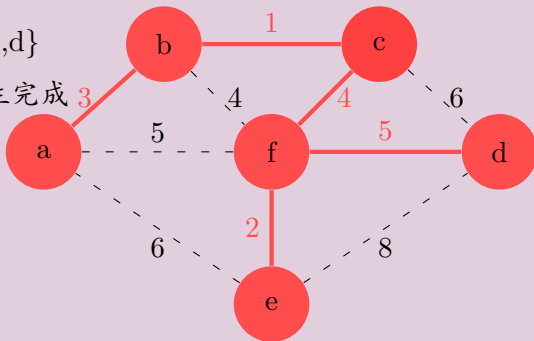
总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续5)

$$V - V_T = \emptyset$$

$$V_T = \{a, b, c, f, e, d\}$$

最小生成树产生完成





Prim算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法核心

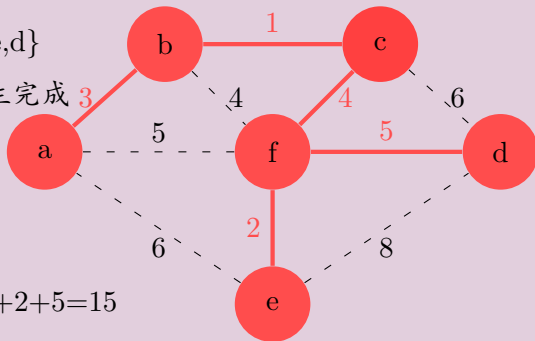
总是找出权重最小的边(a,b), 其中 $a \in V_T$, $b \in V - V_T$

Prim算法实例(续5)

$$V - V_T = \emptyset$$

$$V_T = \{a, b, c, f, e, d\}$$

最小生成树产生完成



$$w(T) = 3 + 1 + 4 + 2 + 5 = 15$$



Prim算法实例总结

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法实例总结

- ① 从图 G 中任意顶点 v_0 开始产生最小生成树
 - 顶点集 V 被划分为两个集合: $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树:总是找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$. 将 b 加入到 V_T 中, 并从 $V - V_T$ 中删除
- ③ 反复执行第2步, 直至集合 $V - V_T$ 为空。



Prim算法实例总结

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法实例总结

- ① 从图 G 中任意顶点 v_0 开始产生最小生成树
 - 顶点集 V 被划分为两个集合: $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树:总是找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$. 将 b 加入到 V_T 中, 并从 $V - V_T$ 中删除
- ③ 反复执行第2步, 直至集合 $V - V_T$ 为空。



Prim算法实例总结

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法实例总结

- ① 从图 G 中任意顶点 v_0 开始产生最小生成树
 - 顶点集 V 被划分为两个集合: $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树:总是找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$. 将 b 加入到 V_T 中, 并从 $V - V_T$ 中删除
- ③ 反复执行第2步, 直至集合 $V - V_T$ 为空。



Prim算法实例总结

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法实例总结

- ① 从图 G 中任意顶点 v_0 开始产生最小生成树
 - 顶点集 V 被划分为两个集合: $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树:总是找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$. 将 b 加入到 V_T 中, 并从 $V - V_T$ 中删除
- ③ 反复执行第2步, 直至集合 $V - V_T$ 为空。



Prim算法实例总结

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法实例总结

- ① 从图 G 中任意顶点 v_0 开始产生最小生成树
 - 顶点集 V 被划分为两个集合: $V_T = \{v_0\}$ 和 $V - V_T$
 - V_T 表示生成树的顶点集
- ② 产生最小生成树:总是找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$. 将 b 加入到 V_T 中, 并从 $V - V_T$ 中删除
- ③ 反复执行第2步, 直至集合 $V - V_T$ 为空。



Prim算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

Prim(G)

//构造最小生成树的Prim算法

//输入：加权连通图 $G=\langle V,E \rangle$

//输出：组成G的最小生成树的边集 E_T

1: $V_T = \{v_0\}$

2: $E_T = \emptyset$

3: **for** $i = 1$ to $|V| - 1$ **do**

4: 找出权重最小的边 (a,b) , 其中 $a \in V_T, b \in V - V_T$

5: $V_T = V_T \cup \{b\}$

6: $V - V_T = V - V_T - \{b\}$

7: $E_T = E_T \cup \{(a,b)\}$

8: **end for**

9: return E_T



Prim算法的时间开销分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法的时间效率分析

```
1: ...  
2: for  $i = 1$  to  $|V| - 1$  do  
3:   找出权重最小的边 $(a,b)$ , 其中 $a \in V_T$ ,  $b \in V - V_T$   
4:   ...  
5: end for
```

- 算法时间复杂度主要由2-3行代码决定！
- FOR循环执行 $|V|-1$ 次
- 确定算法的时间复杂度的核心是第3行代码
- 第3行代码的实现方式决定了它自身的复杂度



Prim算法的时间开销分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法的时间效率分析

```
1: ...  
2: for  $i = 1$  to  $|V| - 1$  do  
3:   找出权重最小的边 $(a,b)$ , 其中 $a \in V_T, b \in V - V_T$   
4:   ...  
5: end for
```

- 算法时间复杂度主要由2-3行代码决定！
- FOR循环执行 $|V|-1$ 次
- 确定算法的时间复杂度的核心是第3行代码
- 第3行代码的实现方式决定了它自身的复杂度



Prim算法的时间开销分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法的时间效率分析

- 1: ...
 - 2: **for** $i = 1$ to $|V| - 1$ **do**
 - 3: 找出权重最小的边 (a,b) , 其中 $a \in V_T, b \in V - V_T$
 - 4: ...
 - 5: **end for**
- 算法时间复杂度主要由2-3行代码决定!
 - FOR循环执行 $|V|-1$ 次
 - 确定算法的时间复杂度的核心是第3行代码
 - 第3行代码的实现方式决定了它自身的复杂度



Prim算法的时间开销分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法的时间效率分析

- 1: ...
 - 2: **for** $i = 1$ **to** $|V| - 1$ **do**
 - 3: 找出权重最小的边 (a,b) , 其中 $a \in V_T, b \in V - V_T$
 - 4: ...
 - 5: **end for**
- 算法时间复杂度主要由2-3行代码决定!
 - FOR循环执行 $|V|-1$ 次
 - 确定算法的时间复杂度的核心是第3行代码
 - 第3行代码的实现方式决定了它自身的复杂度



Prim算法效率分析 - 无序数组实现优先队列

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

1. 无序数组实现优先队列

图是由权重矩阵来表示，而优先队列由一个无序数组来实现



Prim算法效率分析 - 无序数组实现优先队列

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

1. 无序数组实现优先队列

图是由权重矩阵来表示，而优先队列由一个无序数组来实现

复杂度分析

- 第三行代码用数组来实现，每次从数组中选出最小的权值，复杂度为 $O(|V|)$ ；
- 由于for循环，一共需要向 V_T 中加入 $|V| - 1$ 个节点，总的复杂度为 $O(|V|^2)$ ；



2.最小堆实现优先队列

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

时间效率分析

- 最坏情况下，每一层的节点的键都会交换到最底层；
- 移动一次需要两次比较：是否执行交换和找出最大的子女；
- 第 i 层的每个节点（一共有 2^i 个）需要 $2(h - i)$ 次键值比较；
- $C_{worst}(n) = \sum_{i=0}^{\log_2 n - 1} 2^i * 2(\log_2 n - i) = 2(n - \log_2(n + 1))$

堆的操作

插入操作：与堆构造堆相似，最坏情况下的复杂度为 $O(\log_2 n)$ （树的高度）

删除操作：

- 要删除的节点和堆的最后一个节点交换，然后删除；
- “堆化”（自底向上构造堆步骤2-3）删除后的树；



Prim算法的时间开销分析(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

基于最小堆的Prim算法的时间效率分析

使用最小堆存储 $V - V_T$ ，其中键值计算如下：

- 如果 $V - V_T$ 中的顶点与 V_T 相邻，则顶点键值为该顶点离 V_T 的最短边的权重
- 如果 $V - V_T$ 中的顶点与 V_T 不相邻，则顶点键值为 ∞
- 堆的根即为当前 $V - V_T$ 离 V_T 最近的顶点！



Prim算法的时间开销分析(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

基于最小堆的Prim算法的时间效率分析

使用最小堆存储 $V - V_T$ ，其中键值计算如下：

- 如果 $V - V_T$ 中的顶点与 V_T 相邻，则顶点键值为该顶点离 V_T 的最短边的权重
- 如果 $V - V_T$ 中的顶点与 V_T 不相邻，则顶点键值为 ∞
- 堆的根即为当前 $V - V_T$ 离 V_T 最近的顶点！



Prim算法的时间开销分析(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

基于最小堆的Prim算法的时间效率分析

使用最小堆存储 $V - V_T$ ，其中键值计算如下：

- 如果 $V - V_T$ 中的顶点与 V_T 相邻，则顶点键值为该顶点离 V_T 的最短边的权重
- 如果 $V - V_T$ 中的顶点与 V_T 不相邻，则顶点键值为 ∞
- 堆的根即为当前 $V - V_T$ 离 V_T 最近的顶点！



Prim算法的时间开销分析(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

基于最小堆的Prim算法的时间效率分析

使用最小堆存储 $V - V_T$ ，其中键值计算如下：

- 如果 $V - V_T$ 中的顶点与 V_T 相邻，则顶点键值为该顶点离 V_T 的最短边的权重
- 如果 $V - V_T$ 中的顶点与 V_T 不相邻，则顶点键值为 ∞
- 堆的根即为当前 $V - V_T$ 离 V_T 最近的顶点！

找出权重最小的边 (a,b) ，其中 $a \in V_T$ ， $b \in V - V_T$ 的实现如下：

- 1: $u = \text{EXTRACT-HEAP}(V - V_T)$
- 2: **for** v is adjacent to u **do**
- 3: **if** $v \in V - V_T$ and $w(u,v) < \text{key}[v]$ **then**
- 4: $\text{key}[v] = w(u,v)$
- 5: **end if**
- 6: **end for**



Prim算法的时间开销分析(续2)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

```
for  $i = 1$  to  $|V| - 1$  do
```

找出权重最小的边 (a,b) , 其中 $a \in V_T$, $b \in V - V_T$

```
end for
```

上面的代码可以等价转换如下:

```
1: while  $V - V_T \neq \emptyset$  do
```

```
2:    $u = \text{EXTRACT-HEAP}(V - V_T)$ 
```

```
3:   for  $v$  is adjacent to  $u$  do
```

```
4:     if  $v \in V - V_T$  and  $w(u,v) < \text{key}[v]$  then
```

```
5:        $\text{key}[v] = w(u,v)$ 
```

```
6:     end if
```

```
7:   end for
```

```
8: end while
```

实现第2行代码的复杂度为 $O(\log |V|)$



Prim算法的时间开销分析(续3)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

```
1: while  $V - V_T \neq \emptyset$  do  
2:    $u = \text{EXTRACT-HEAP}(V - V_T)$   
3:   for  $v$  is adjacent to  $u$  do  
4:     if  $v \in V - V_T$  and  $w(u,v) < \text{key}[v]$  then  
5:        $\text{key}[v] = w(u,v)$   
6:     end if  
7:   end for  
8: end while
```



Prim算法的时间开销分析(续3)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

```
1: while  $V - V_T \neq \emptyset$  do
2:    $u = \text{EXTRACT-HEAP}(V - V_T)$ 
3:   for  $v$  is adjacent to  $u$  do
4:     if  $v \in V - V_T$  and  $w(u,v) < \text{key}[v]$  then
5:        $\text{key}[v] = w(u,v)$ 
6:     end if
7:   end for
8: end while
```

基于最小堆的Prim算法的时间效率分析

- 第2行代码的时间复杂度为 $O(\log |V|)$
- 1-2行代码的时间复杂度为 $(|V| - 1)O(\log |V|)$
- 1,3行代码(两次循环)总共执行多少次? $2|E|$ 次
- 每一次FOR循环, $\text{key}[v] = w(u,v)$ 可能被执行, 这会触发堆的重构, 其时间复杂度为 $O(\log |V|)$
- 算法时间复杂度为 $(|V|-1+|E|) O(\log |V|)$



Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算
法

最小生成树

加权连通图 $G=\langle V,E \rangle$ 的一棵**最小生成树**是图的一棵权重最小的生成树



Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

最小生成树

加权连通图 $G=\langle V,E \rangle$ 的一棵**最小生成树**是图的一棵权重最小的生成树

Kruskal算法的思想

总是选取 E 中最短的边来构成树。是否能够产生最小生成树，试一试！

- 图中边集 E 按照权重的非递减顺序进行排序
- 总是选取边值最小的边构成树
- 所选的边不能与已产生的树构成回路！



Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

最小生成树

加权连通图 $G=\langle V, E \rangle$ 的一棵**最小生成树**是图的一棵权重最小的生成树

Kruskal算法的思想

总是选取 E 中最短的边来构成树。是否能够产生最小生成树，试一试！

- 图中边集 E 按照权重的非递减顺序进行排序
- 总是选取边值最小的边构成树
- 所选的边不能与已产生的树构成回路！

终止条件

- 图的最小生成树必须包含所有顶点 V ，它的边数只能为 $|V|-1$
- 终止条件：选择 $|V|-1$ 条边后终止，最小生成树生成！



Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

最小生成树

加权连通图 $G=\langle V, E \rangle$ 的一棵**最小生成树**是图的一棵权重最小的生成树

Kruskal算法的思想

总是选取 E 中最短的边来构成树。是否能够产生最小生成树，试一试！

- 图中边集 E 按照权重的非递减顺序进行排序
- 总是选取边值最小的边构成树
- 所选的边不能与已产生的树构成回路！

终止条件

- 图的最小生成树必须包含所有顶点 V ，它的边数只能为 $|V|-1$
- 终止条件：选择 $|V|-1$ 条边后终止，最小生成树生成！



Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

//构造最小生成树的Kruskal算法

//输入：加权连通图 $G=\langle V,E \rangle$

//输出： E_T

- 1: 按照边的权重非递减顺序对集合 E 排序
- 2: $E_T = \emptyset$; encounter = 0;
- 3: **while** encounter < $|V|-1$ **do**
- 4: **if** $E_T \cup (a,b)$ 无回路 **then**
- 5: $E_T = E_T \cup (a,b)$;
- 6: encounter = encounter + 1;
- 7: **end if**
- 8: **end while**
- 9: return E_T



Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

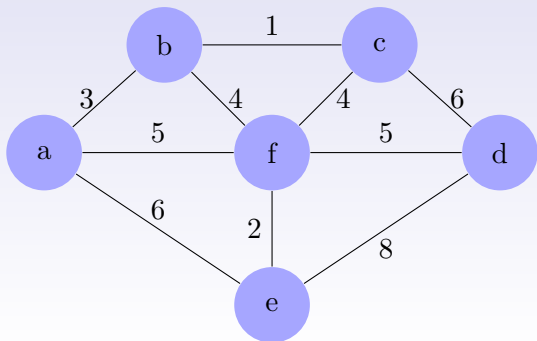
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5), (a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

- 第一步对E进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4), (c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), $\text{encounter} = 0$;
- 依次选择最小边构成生成树, 但所选边不能构成回路





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

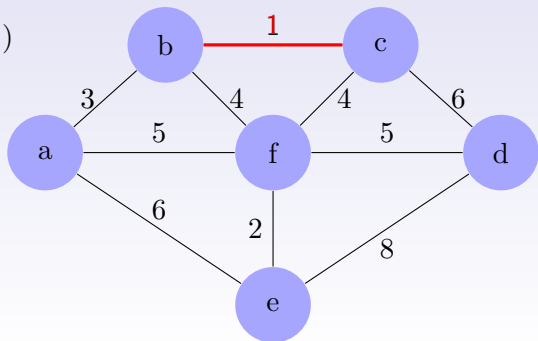
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5), (a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

- 第一步对 E 进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4), (c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), $\text{encounter} = 0$;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

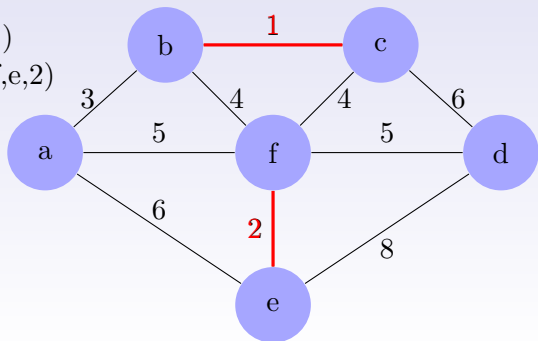
最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5), (a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

- 第一步对 E 进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4), (c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), $\text{encounter} = 0$;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$
$$E_T = (b,c,1),(f,e,2)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

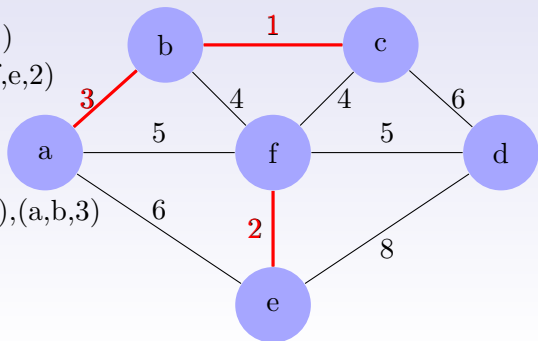
一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5), (a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

- 第一步对E进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4), (c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), encounter = 0;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$

$$E_T = (b,c,1),(f,e,2)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5),(a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

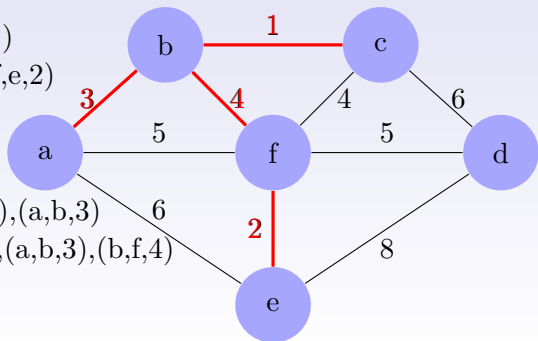
- 第一步对E进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4),(c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), $\text{encounter} = 0$;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$

$$E_T = (b,c,1),(f,e,2)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3),(b,f,4)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E\rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5),(a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

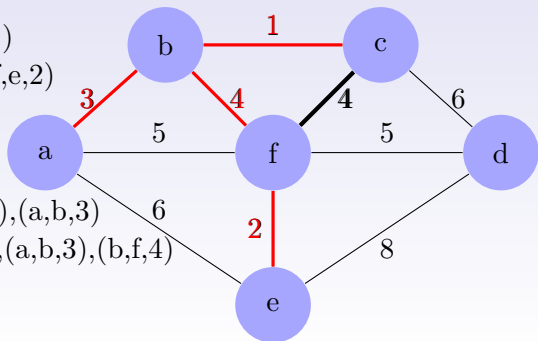
- 第一步对E进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4),(c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), encounter = 0;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$

$$E_T = (b,c,1),(f,e,2)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3),(b,f,4)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E \rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5),(a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

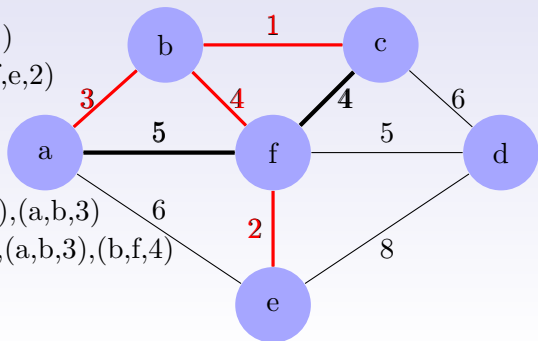
- 第一步对 E 进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4),(c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), $\text{encounter} = 0$;
- 依次选择最小边构成生成树, 但所选边不能构成回路

$$E_T = (b,c,1)$$

$$E_T = (b,c,1),(f,e,2)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3),(b,f,4)$$





Kruskal算法实例

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

一个图 $G=\langle V,E \rangle$ 如下所示, $V=\{a,b,c,d,e,f\}$, $E=\{(a,b,3),(a,f,5),(a,e,6),(b,c,1),(b,f,4),(c,d,6),(c,f,4),(d,e,8),(d,f,5),(f,e,2)\}$

- 第一步对E进行排序: $E=\{(b,c,1),(f,e,2),(a,b,3),(b,f,4),(c,f,4),(a,f,5),(d,f,5),(c,d,6),(a,e,6),(d,e,8)\}$
- $E_T=\emptyset$ (图中红线组成), encounter = 0;
- 依次选择最小边构成生成树, 但所选边不能构成回路

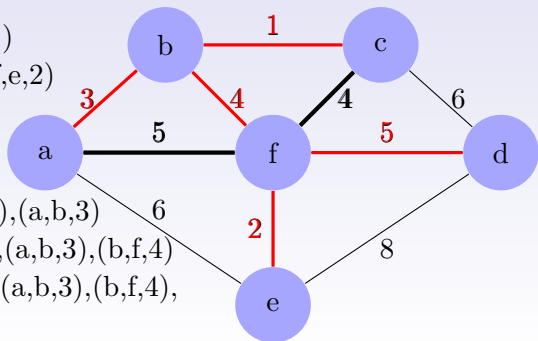
$$E_T = (b,c,1)$$

$$E_T = (b,c,1),(f,e,2)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3),(b,f,4)$$

$$E_T = (b,c,1),(f,e,2),(a,b,3),(b,f,4), \\ (f,d,6)$$





Kruskal算法实例(续)

贪心算法

讲授者 王爱娟

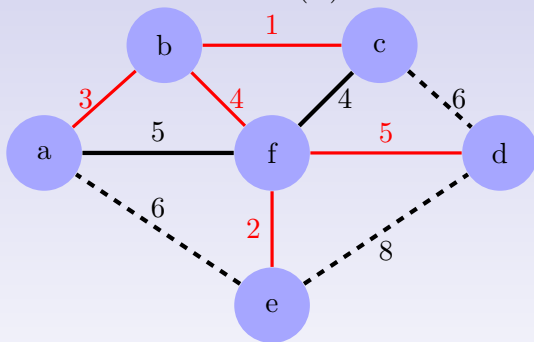
目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

最小生成树的权值为: $w(T)=1+2+3+4+5=15$



Kruskal算法的关键

如何确保新加入的边与已产生的树不构成回路?



判断新加入边是否构成回路

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$$A = \{a,b,c,d,e,f\} \xrightarrow{\text{partition}} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$

union操作和find操作

- $\text{union}(a,b) = \{a,b\}$
- $\text{union}(a,c) = \{a,b,c\}$
- $\text{union}(d,e) = \{d,e\}$
- $\text{union}(a,d) = \{a,b,c,d,e\}$
- 两个元素的union = 包含该元素的集合的并



判断新加入边是否构成回路

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$$A = \{a,b,c,d,e,f\} \xrightarrow{\text{partition}} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$

union操作和find操作

- $\text{union}(a,b) = \{a,b\}$
- $\text{union}(a,c) = \{a,b,c\}$
- $\text{union}(d,e) = \{d,e\}$
- $\text{union}(a,d) = \{a,b,c,d,e\}$
- 两个元素的union = 包含该元素的集合的并

find操作

- $\Rightarrow \text{find}(a) = \text{find}(b) = \{a,b\}$
- $\text{find}(a) = \text{find}(b) = \text{find}(c) = \{a,b,c\}$
- $\text{find}(d) = \text{find}(e) = \{d,e\}$
- $\text{find}(a) = \text{find}(d) = \{a,b,c,d,e\}$
- find 一个元素 = 返回包含该元素的集合



判断新加入边是否构成回路

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$$A = \{a,b,c,d,e,f\} \xrightarrow{\text{partition}} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$

union操作和find操作

- $\text{union}(a,b) = \{a,b\}$
- $\text{union}(a,c) = \{a,b,c\}$
- $\text{union}(d,e) = \{d,e\}$
- $\text{union}(a,d) = \{a,b,c,d,e\}$
- 两个元素的union = 包含该元素的集合的并

find操作

- $\Rightarrow \text{find}(a) = \text{find}(b) = \{a,b\}$
- $\text{find}(a) = \text{find}(b) = \text{find}(c) = \{a,b,c\}$
- $\text{find}(d) = \text{find}(e) = \{d,e\}$
- $\text{find}(a) = \text{find}(d) = \{a,b,c,d,e\}$
- find一个元素 = 返回包含该元素的集合

union和find操作能否用于判断回路？



应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

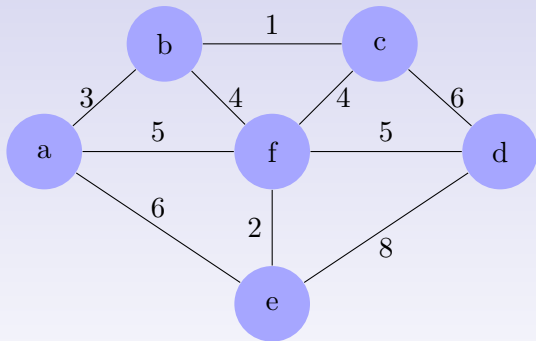
目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$
$$E_T = \{\emptyset\}, V_T = \{\emptyset\}$$





应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

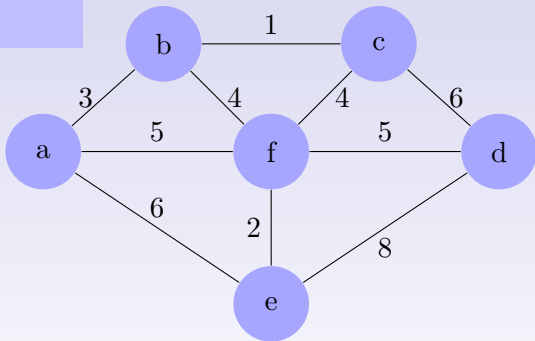
最小生成树
之Kruskal算法

$$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$

$$E_T = \{\emptyset\}, V_T = \{\emptyset\}$$

$$\text{find}(b) = \{b\}$$

$$\text{find}(c) = \{c\}$$





应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

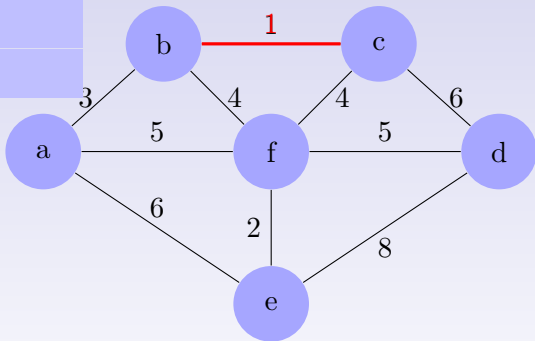
$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$

$E_T = \{\emptyset\}, V_T = \{\emptyset\}$

$\text{find}(b) = \{b\}$

$\text{find}(c) = \{c\}$

$\text{union}(b, c) = \{b, c\}$





应用union和find操作与Kruskal算法

贪心算法

讲者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$$

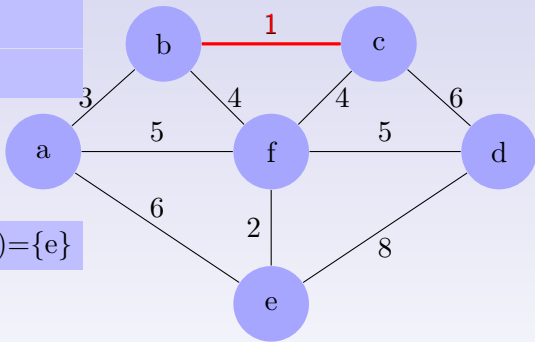
$$E_T = \{\emptyset\}, V_T = \{\emptyset\}$$

$$\text{find}(b) = \{b\}$$

$$\text{find}(c) = \{c\}$$

$$\text{union}(b, c) = \{b, c\}$$

$$\text{find}(f) = \{f\} \quad \text{find}(e) = \{e\}$$





应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$

$E_T = \{\emptyset\}, V_T = \{\emptyset\}$

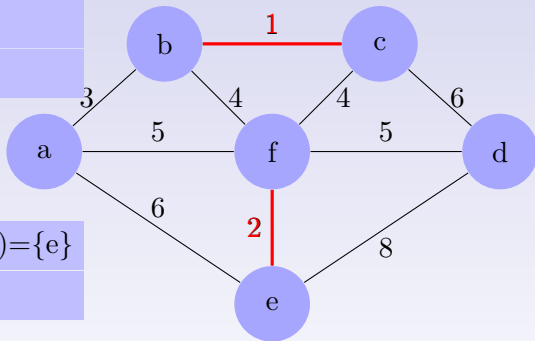
$\text{find}(b) = \{b\}$

$\text{find}(c) = \{c\}$

$\text{union}(b, c) = \{b, c\}$

$\text{find}(f) = \{f\} \quad \text{find}(e) = \{e\}$

$\text{union}(e, f) = \{e, f\}$





应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$

$E_T = \{\emptyset\}, V_T = \{\emptyset\}$

$\text{find}(b) = \{b\}$

$\text{find}(c) = \{c\}$

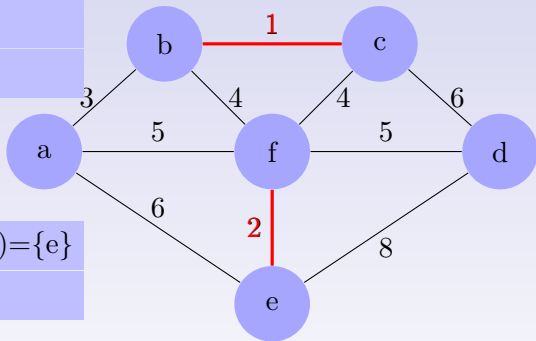
$\text{union}(b, c) = \{b, c\}$

$\text{find}(f) = \{f\} \quad \text{find}(e) = \{e\}$

$\text{union}(e, f) = \{e, f\}$

$\text{find}(a) = \{a\}$

$\text{find}(b) = \{b, c\}$





应用union和find操作与Kruskal算法

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$V = \{a, b, c, d, e, f\} \xrightarrow{part} \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}$

$E_T = \{\emptyset\}, V_T = \{\emptyset\}$

$\text{find}(b) = \{b\}$

$\text{find}(c) = \{c\}$

$\text{union}(b, c) = \{b, c\}$

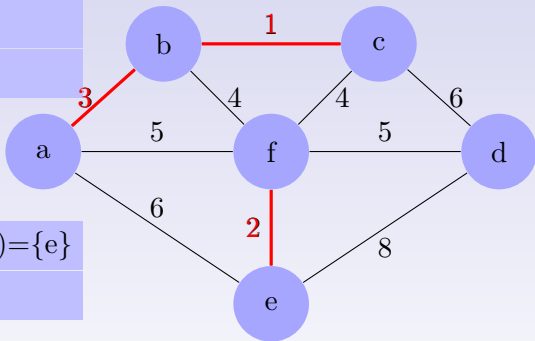
$\text{find}(f) = \{f\} \quad \text{find}(e) = \{e\}$

$\text{union}(e, f) = \{e, f\}$

$\text{find}(a) = \{a\}$

$\text{find}(b) = \{b, c\}$

$\text{union}(a, b) = \{a, b, c\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

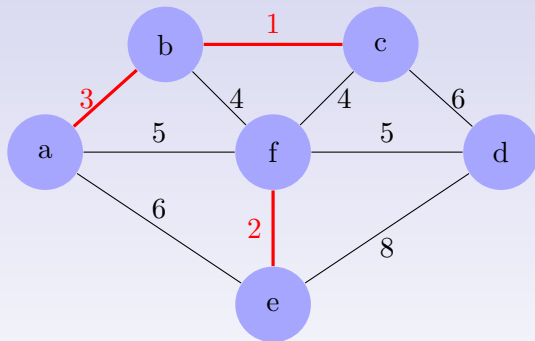
贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

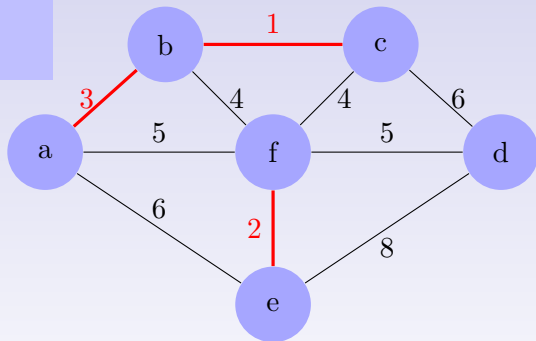
目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$
 $E_T = \{(a, b), (b, c), (e, f)\},$
 $\text{find}(b) = \{a, b, c\}$
 $\text{find}(f) = \{e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

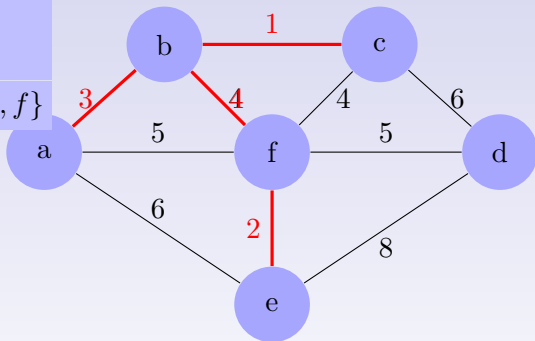
目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$
 $E_T = \{(a, b), (b, c), (e, f)\},$
 $\text{find}(b) = \{a, b, c\}$
 $\text{find}(f) = \{e, f\}$
 $\text{union}(b, f) = \{a, b, c, e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

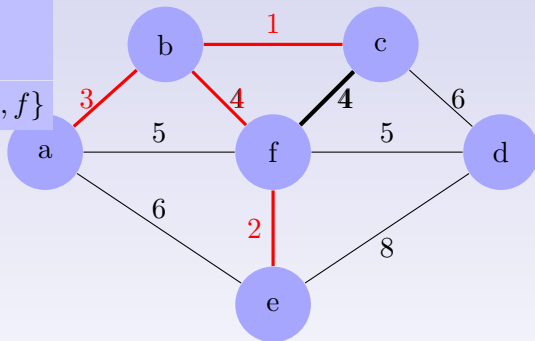
$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

$\text{union}(b, f) = \{a, b, c, e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

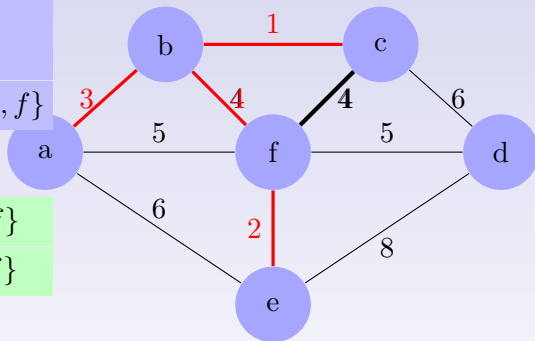
$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

$\text{union}(b, f) = \{a, b, c, e, f\}$

$\text{find}(c) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

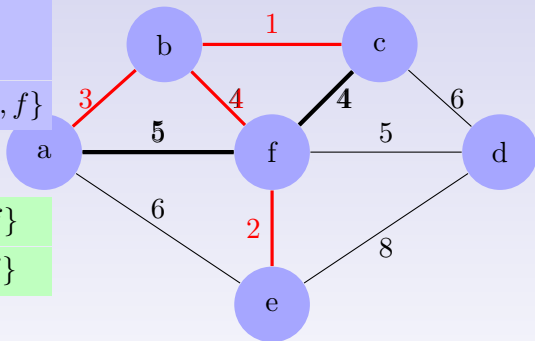
$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

$\text{union}(b, f) = \{a, b, c, e, f\}$

$\text{find}(c) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

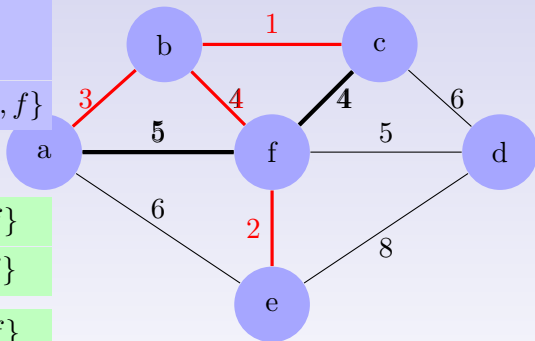
$\text{union}(b, f) = \{a, b, c, e, f\}$

$\text{find}(c) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$

$\text{find}(a) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$





应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

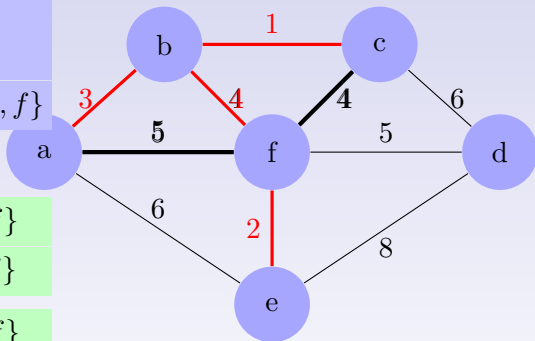
$\text{union}(b, f) = \{a, b, c, e, f\}$

$\text{find}(c) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$

$\text{find}(a) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$



$\text{find}(f) = \{a, b, c, e, f\}$

$\text{find}(d) = \{d\}$

$\text{union}(f, d) = \{a, b, c, d, e, f\}$



应用union和find操作与Kruskal算法(续1)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

$\{a, b, c\}, \{d\}, \{e, f\}$

$E_T = \{(a, b), (b, c), (e, f)\},$

$\text{find}(b) = \{a, b, c\}$

$\text{find}(f) = \{e, f\}$

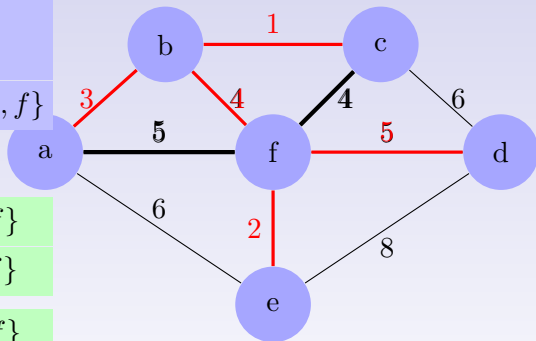
$\text{union}(b, f) = \{a, b, c, e, f\}$

$\text{find}(c) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$

$\text{find}(a) = \{a, b, c, e, f\}$

$\text{find}(f) = \{a, b, c, e, f\}$



$\text{find}(f) = \{a, b, c, e, f\}$

$\text{find}(d) = \{d\}$

$\text{union}(f, d) = \{a, b, c, d, e, f\}$



应用union和find操作与Kruskal算法(续2)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

应用union和find判断回路

若边(a,b)被加入到生成树。如果 $\text{find}(a)=\text{find}(b)$ 则表明该边会导致回路，不能添加；否则，则可以添加。



应用union和find操作与Kruskal算法(续2)

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

应用union和find判断回路

若边(a,b)被加入到生成树。如果 $\text{find}(a)=\text{find}(b)$ 则表明该边会导致回路，不能添加；否则，则可以添加。

Kruskal(G)

//构造最小生成树的Kruskal算法

//输入：加权连通图 $G=\langle V,E \rangle$

//输出： E_T

- 1: 按照边的权重非递减顺序对集合E排序
- 2: $E_T = \emptyset$; encounter = 0;
- 3: **while** encounter < $|V|-1$ **do**
- 4: 取下一条最短边(a,b)
- 5: **if** $\text{find}(a) \neq \text{find}(b)$ **then**
- 6: $E_T = E_T \cup (a,b)$; union(a,b); encounter++;
- 7: **end if**
- 8: **end while**
- 9: **return** E_T



Kruskal算法的时间复杂度分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

- 1: 按照边的权重非递减顺序对集合E排序
- 2: **while** encounter $< |V|-1$ **do**
- 3: **if** find(a) \neq find(b) **then**
- 4: union(a,b);
- 5: **end if**
- 6: **end while**
- 7: return E_T



Kruskal算法的时间复杂度分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

```
1: 按照边的权重非递减顺序对集合E排序
2: while encounter <  $|V|-1$  do
3:   if find(a)  $\neq$  find(b) then
4:     union(a,b);
5:   end if
6: end while
7: return  $E_T$ 
```

Kruskal(G)-快速求并(P254)

- 第1行代码的时间复杂度为 $O(|E| \log |E|)$ (快速排序)
- 一次union操作的时间复杂度为 $O(1)$
- 2-3行代码的 m 次find操作的时间复杂度为 $O(m \log |V|)$
- 总的时间复杂度为 $O(|V|)+O(|E| \log |E|) + O(m \log |V|)$



Kruskal算法的时间复杂度分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

```
1: 按照边的权重非递减顺序对集合E排序
2: while encounter < |V|-1 do
3:   if find(a) ≠ find(b) then
4:     union(a,b);
5:   end if
6: end while
7: return  $E_T$ 
```

Kruskal(G)-快速求并(P254)

- 第1行代码的时间复杂度为 $O(|E| \log |E|)$ (快速排序)
- 一次union操作的时间复杂度为 $O(1)$
- 2-3行代码的 m 次find操作的时间复杂度为 $O(m \log |V|)$
- 总的时间复杂度为 $O(|V|) + O(|E| \log |E|) + O(m \log |V|)$



Kruskal算法的时间复杂度分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

```
1: 按照边的权重非递减顺序对集合E排序
2: while encounter <  $|V|-1$  do
3:   if find(a)  $\neq$  find(b) then
4:     union(a,b);
5:   end if
6: end while
7: return  $E_T$ 
```

Kruskal(G)-快速求并(P254)

- 第1行代码的时间复杂度为 $O(|E| \log |E|)$ (快速排序)
- 一次union操作的时间复杂度为 $O(1)$
- 2-3行代码的 m 次find操作的时间复杂度为 $O(m \log |V|)$
- 总的时间复杂度为 $O(|V|)+O(|E| \log |E|) + O(m \log |V|)$



Kruskal算法的时间复杂度分析

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Kruskal(G)

```
1: 按照边的权重非递减顺序对集合E排序
2: while encounter <  $|V|-1$  do
3:   if find(a)  $\neq$  find(b) then
4:     union(a,b);
5:   end if
6: end while
7: return  $E_T$ 
```

Kruskal(G)-快速求并(P254)

- 第1行代码的时间复杂度为 $O(|E| \log |E|)$ (快速排序)
- 一次union操作的时间复杂度为 $O(1)$
- 2-3行代码的 m 次find操作的时间复杂度为 $O(m \log |V|)$
- 总的时间复杂度为 $O(|V|)+O(|E| \log |E|) + O(m \log |V|)$



Prim算法与Kruskal算法的比较

贪心算法

讲授者 王爱娟

目录

贪心法的思想

最小生成树
之PRIM算法

最小生成树
之Kruskal算法

Prim算法与Kruskal算法的比较

- 贪心法的思想：总是选择当前最优的解
- 不同的贪心选择，导致不同的贪心算法。因此，贪心算法的好坏关键在于正确的贪心选择。
▶ Jump to different greedy choices
- Prim的贪心选择：距离最近的点
- Kruskal的贪心选择：权值最短的边