

Graph

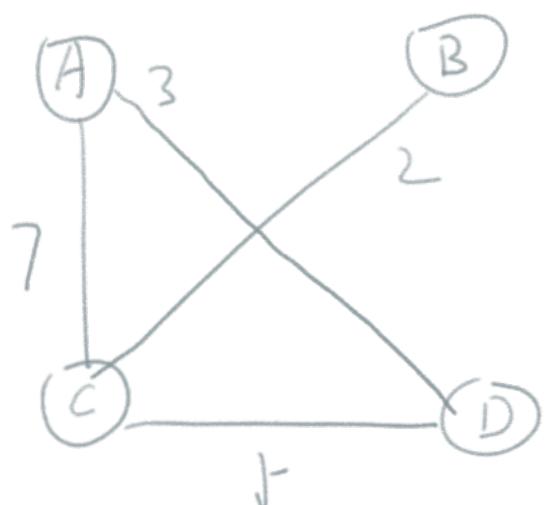
2022年5月23日 17:28

图：

图的两种存储方式：

1. 有向图 $A \rightarrow C$

2. 无向图 $A - C$



邻接表法：

A:		C: $\begin{cases} D \\ \infty \end{cases}$
B:		D: C
C:		A, B, D
D:		A, C

封装数据

邻接矩阵：

	A	B	C	D
A	0	∞	7	3
B	∞	0	2	∞
C	7	2	0	5
D	3	∞	5	0

表达图的方式很多：

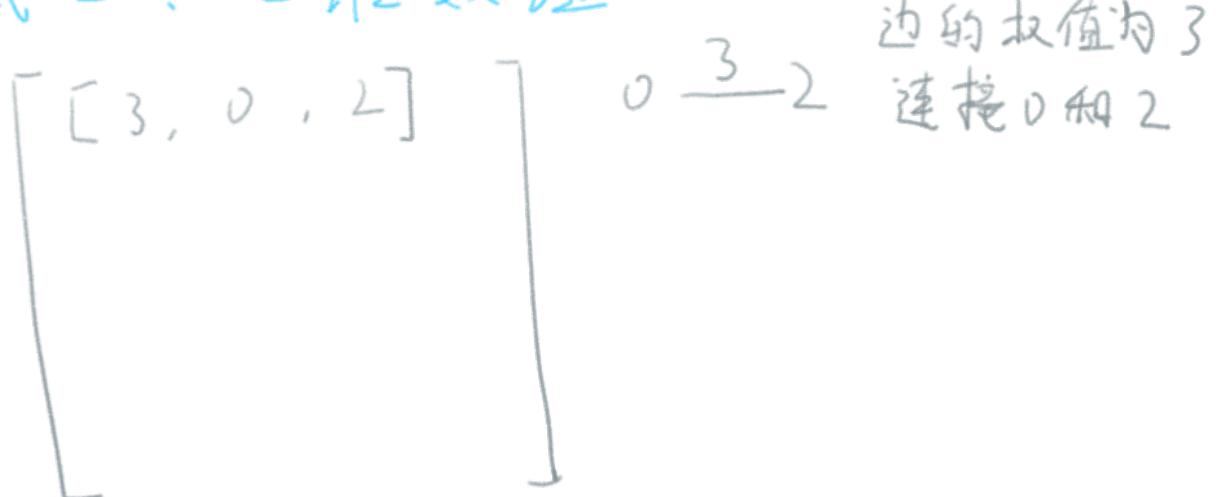
数组可以表达特殊的图：

arr [5 2 2 4 2 1] 终点 指向同一

$\text{arr} [5 \ 2 \ 2 \ 4 \ 2 \ 1]$ 终点 指向同一
0 1 2 3 4 5 ← 城市 方向



方式二：二维数组



图的宽度优先遍历：

1. 利用队列实现

2. 从源节点开始依次按照宽度进队列，然后弹出

3. 每弹出一个点，把所有没有进过该队列的邻接点放入队列

4. 直到队列变空

广度优先遍历：

1. 利用栈实现

2. 从源节点开始把节点按照深度放入栈，然后弹出
3. 每弹出一个点，把该节点下一个没有进过栈的邻接点放入栈
4. 直到栈变空

拓扑排序问题：

比如包之间的依赖关系：



先找入度为0的点：A，并把这个点及其影响擦掉



下一个入度为0的点是B



然后把C擦掉



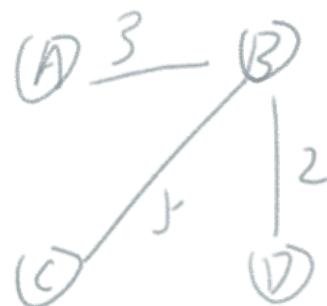
Kruskal 和 Prim 算法：

最小生成树（要求是无向图）

Kruskal 算法
用于生成最小生成树(要求是无向图)



最小生成树



保证所有点都是连通的 同时 权值之和
最小

方法：判断把最小的边 加上是否 会
形成环

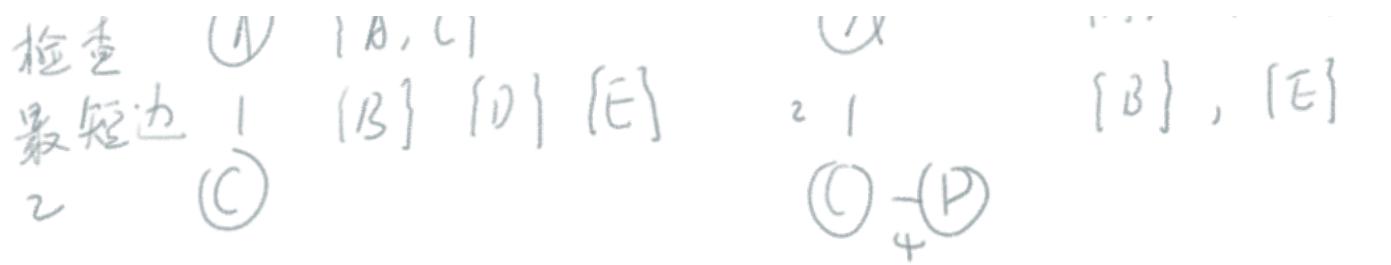


{A} {B} {C}
{D} {E}

如果不在一个集合中
就可以连接

检查
① {A, C}
② {B, D, E}

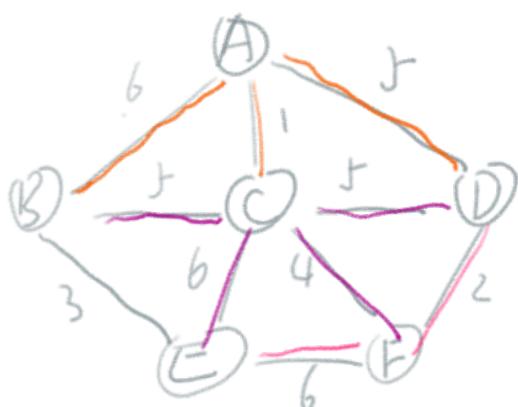
① {A, C, D}
, {B}, {E}



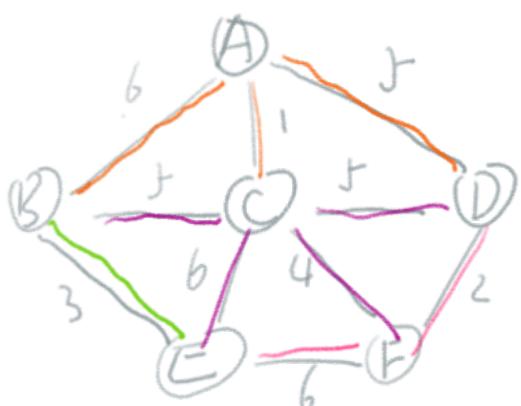
⑪ - ⑫ 在一个集合中 \times
 ⑬ - ⑭ \times
 ⑮ - ⑯ 不在 \checkmark

结果：
 ⑰ $\{A, B, C, D\}$ ⑱ $\{E\}$
 2 ⑲ $\{A, B, C, D\}$ ⑳ $\{E\}$

Prim 算法：



随便选一个点 ①
 在 ② 的边中选最短边
 ③ $\{A, C\}$
 把 ④ 加进来 $\{A, C\}$
 ⑤ 的最短边 ⑥ $\{A, C, F\}$
 加进来 $\{A, C, F\}$
 把 ⑦ $\{D\}$ 加进来、
 $\{A, C, F, D\}$

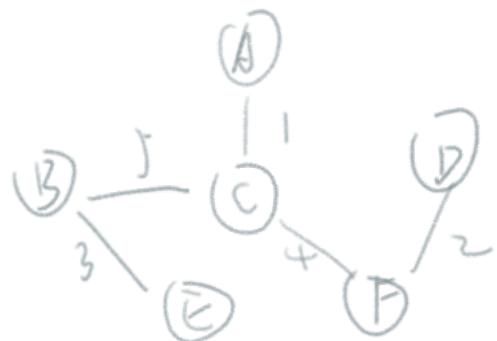


没有新的边被解锁，从
 ⑧ $\{A, C, F, D\}$ ⑨ $\{B\}$ ⑩ $\{D\}$ ⑪ $\{B\}$ ⑫ $\{D\}$
 ⑬ $\{A, C, F, D\}$ ⑭ $\{B\}$ ⑮ $\{D\}$ ⑯ $\{B\}$ ⑰ $\{D\}$

$\textcircled{C} - \textcircled{E}$

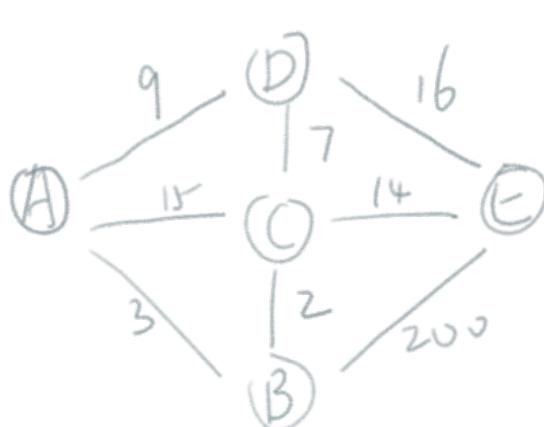
$\textcircled{D} - \textcircled{E}$ 和 $\textcircled{C} - \textcircled{D}$ 的点已经
包含了，所以选 $\textcircled{B} - \textcircled{C}$

加入 $\textcircled{B} - \textcircled{C}$ 在所有被加入的边中 $\textcircled{B} - \textcircled{C}$
最短，所以传递为 {A, C, F, D, B, E}



Dijkstra 算法：

不能有权值为负数的边 和为负数的环
一定要规定出发点，算出出发点到每个点的距离



	B	C	D	E
A	3	—	9	19

初始：

	A	B	C	D	E
A	0	∞	∞	∞	∞

不变了 挑 \textcircled{C}

	A	B	C	D	E
A	0	∞	3	∞	∞

从 \textcircled{A} 开始，有更小值就

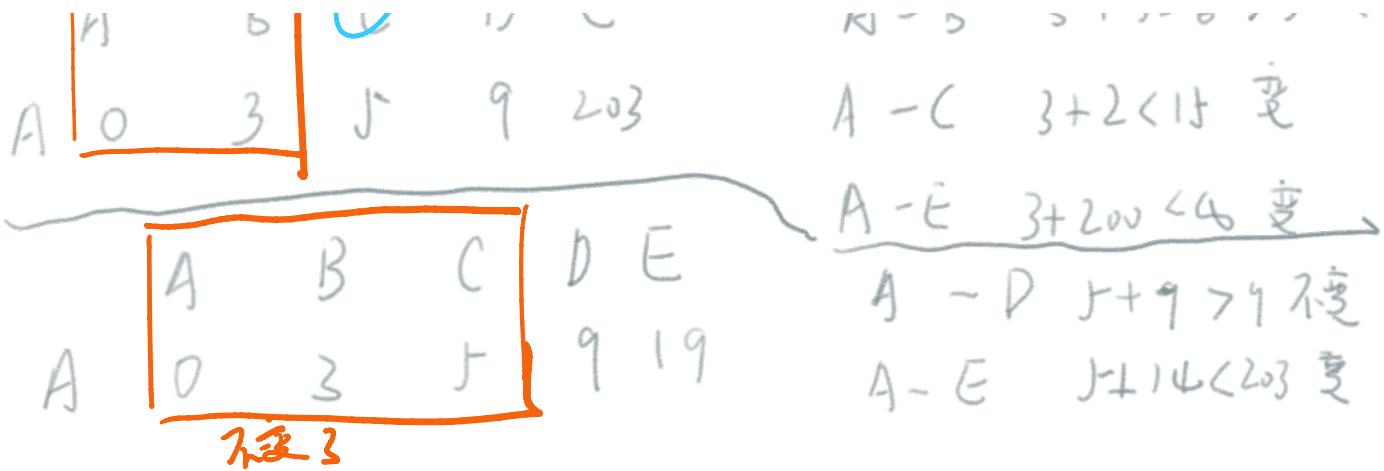
更新

	A	B	C	D	E
A	0	3	15	9	∞

不变了 然后挑 \textcircled{B}

$$A - B: 3 + 3 = 6 > 3 \text{ 不变}$$

$$A - C: 15 + 7 = 22 < 15 \text{ 变}$$



前缀树：

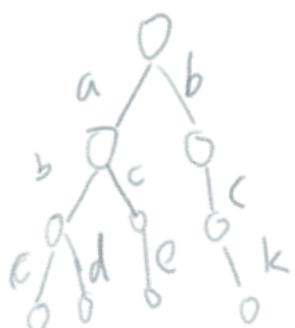
["abc", "bck", "abd", "ace"]

○ 新建空节点，没有到 a 的路，需要建

字符串放在路径上 不在点上

"abd" 有到 ab 的路，可以复用

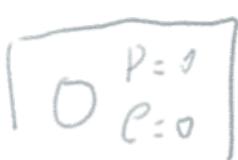
为了实现一些功能，
点中可以包含信息



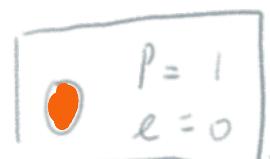
pass 和 end 值

nexts[]: 是否有到 a-z 的路

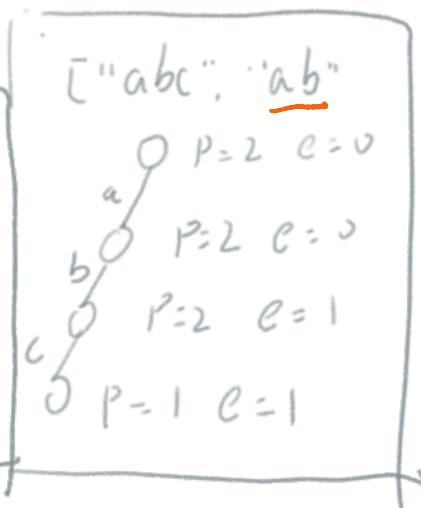
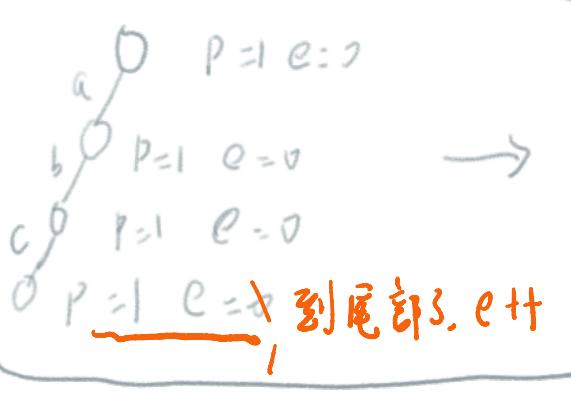
["abc"]



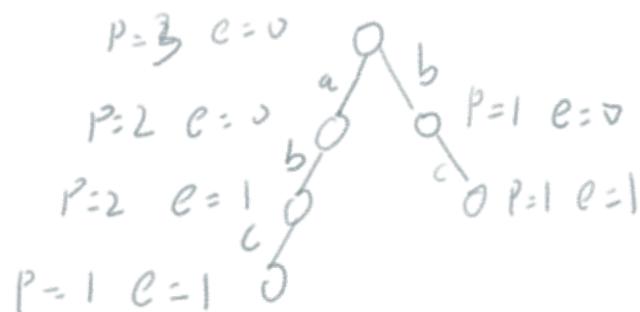
转到头节点



nexts[0] := null
建立起这条路



[“abc”, “ab”, “bc”]



c值作用：查看是否有
加入过“ab”

P值：有多少个是以
“ab”为前缀的