离散数学

Discrete Mathematics

for Computer Science

计算机学院计科系

薛思清 xuesiqing@cug.edu.cn

第13讲 图论 Graph Theory-Introduction(2)

"The origins of graph theory are humble, even frivolous."

——N. Biggs, E. K. Lloyd, and R. J. Wilson (Graph Theory: 1736-1936)

Outline of this lecture

图论基本术语

图基本性质

图的连通性

图的连通性

(1) 通路(walk)-无相同结点? -路/路径(path), 常使用$\Gamma_{1,2}$表示结点v_1到v_2间的路径。

通路-首尾结点相同? -(开/闭)通路;

通路-无相同边? -迹(trail)-首尾结点相同? -回路(circuit) -无相同结点?

-环(cycle/polygon)

——路径长度

(2) 两个结论:

a.存在通路(闭通路)的两结点间必存路(环)

b.任意结点度数>=2的图包含回路

(3) 连通性: 连通/可达(Connected)

(4) (无向)连通图(Connected graph)

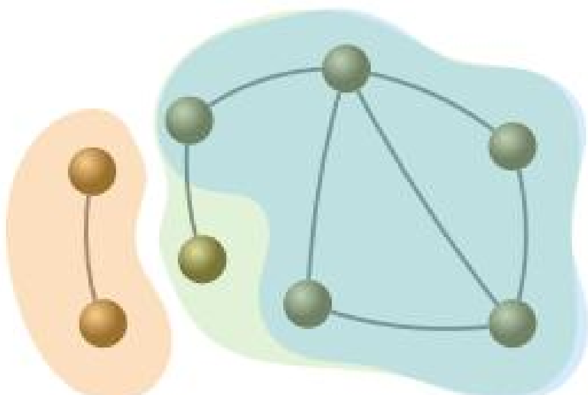
连通关系——划分

(5) (连通)分图 (Connected component)

(6) 从集合角度?

Connectivity, components

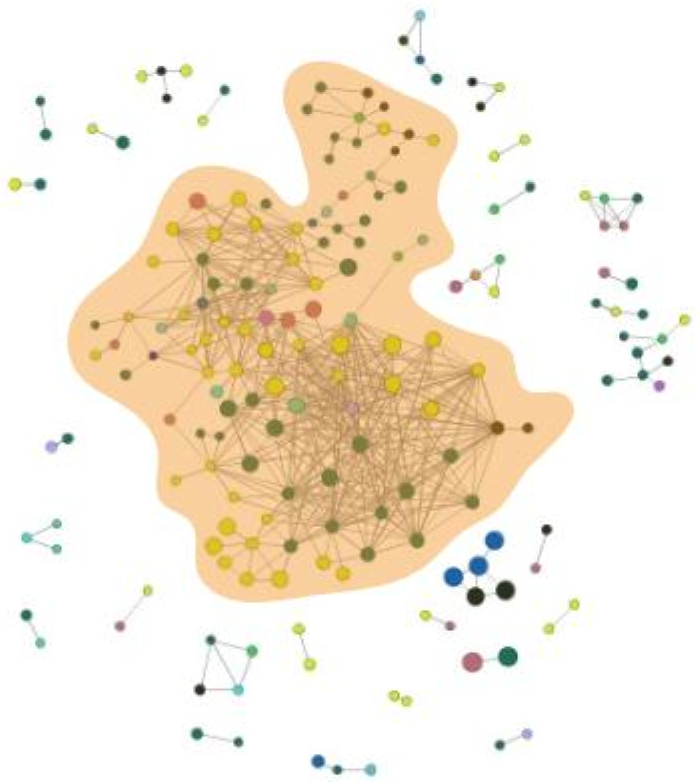
- A subgraph is *connected* if there is a path between every pair of nodes
- A *component* of a graph is a maximal connected subgraph

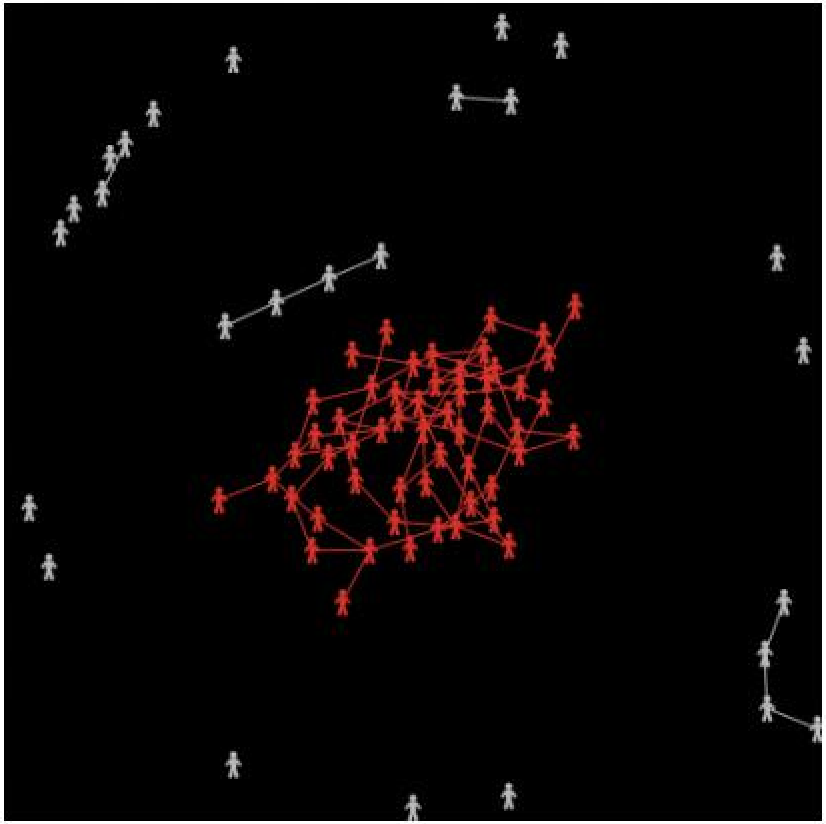


——by Ozalp Babaoglu

Giant components

- If the largest component of a graph contains a significant proportion of all nodes, it is called the *giant component*





——by Ozalp Babaoglu

One example, is a graph formed from a data base of protean interactions where vertices correspond to proteins and edges correspond to pairs of proteins that interact. The graph has 2735 vertices and 3602 edges. The associated graph had the number of components of various sizes shown in Table 1.

SIZE OF COMPONENT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	…	1851
NUMBER OF COMPONENTS	48	179	50	25	14	6	4	6	1	1	1	0	0	0	0	1		1

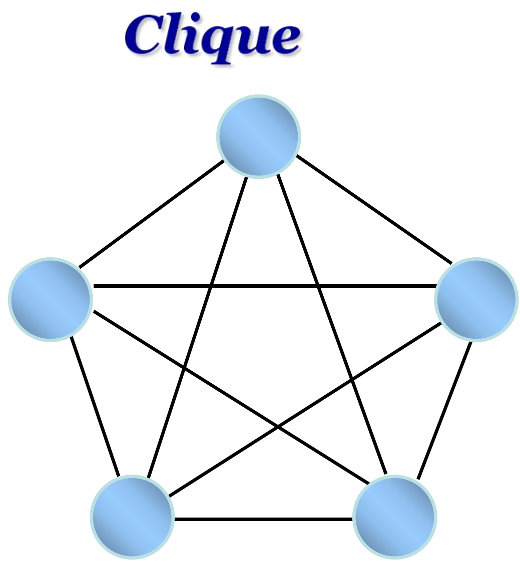
Another example,

Vertices are papers and edges mean that two papers shared an author.

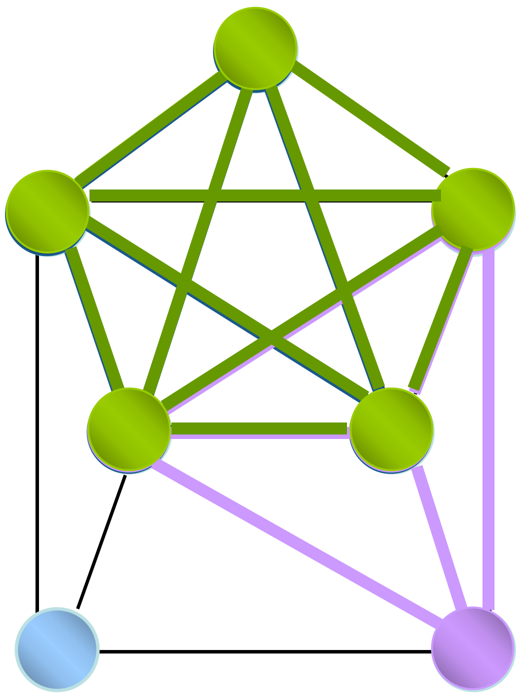
1	2	3	4	5	6	7	8	14	27488
2712	549	129	51	16	12	8	3	1	1

Clique(团)

Each pair of vertices is connected.



$$\boxed{CLIQUE = \{< G, k > | \ G \text{ has a clique of size } k\}}$$



Maximum Clique of Size 5

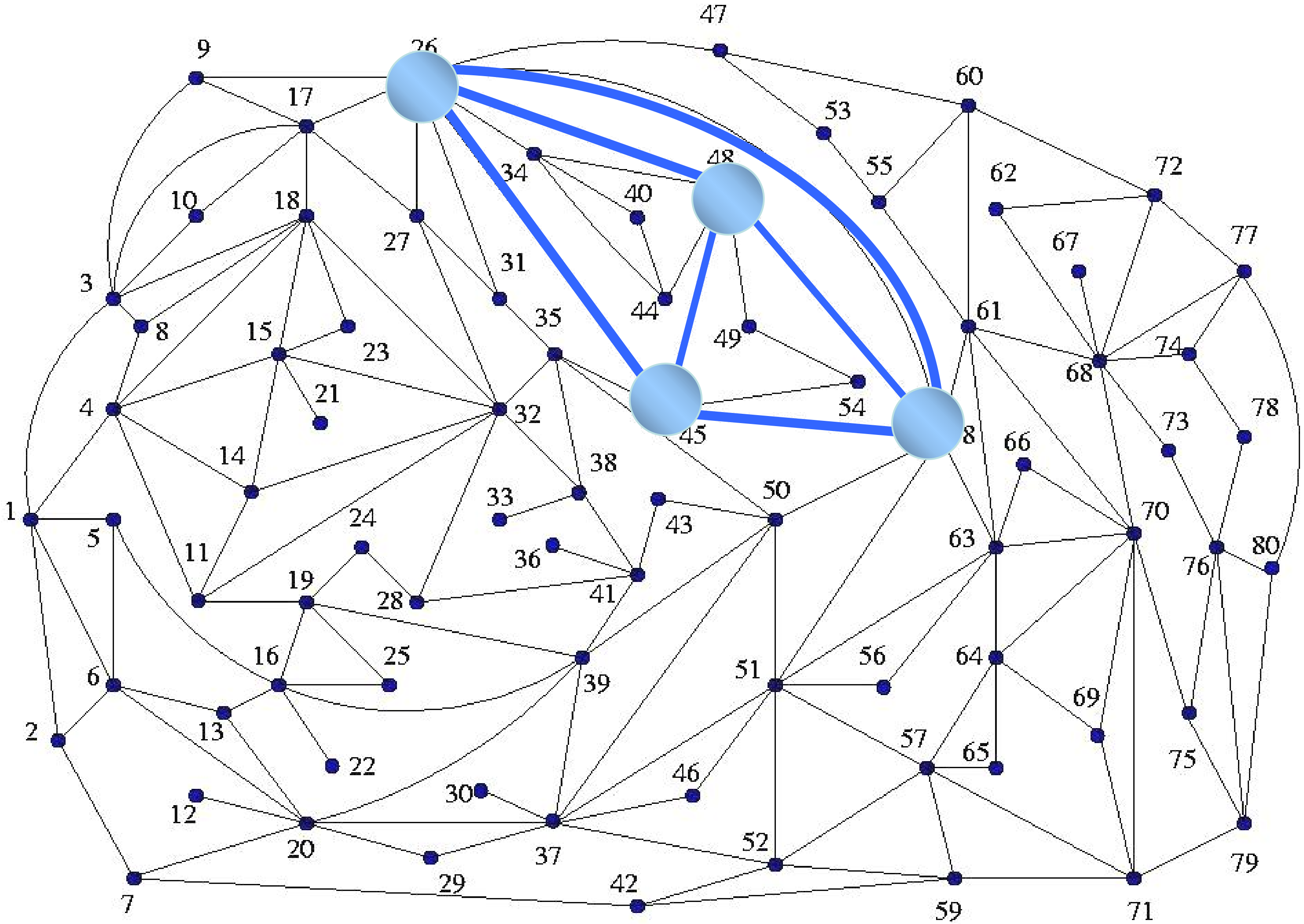
Clique: a complete subgraph

Maximal Clique: a *clique* cannot be enlarged by adding any more vertices

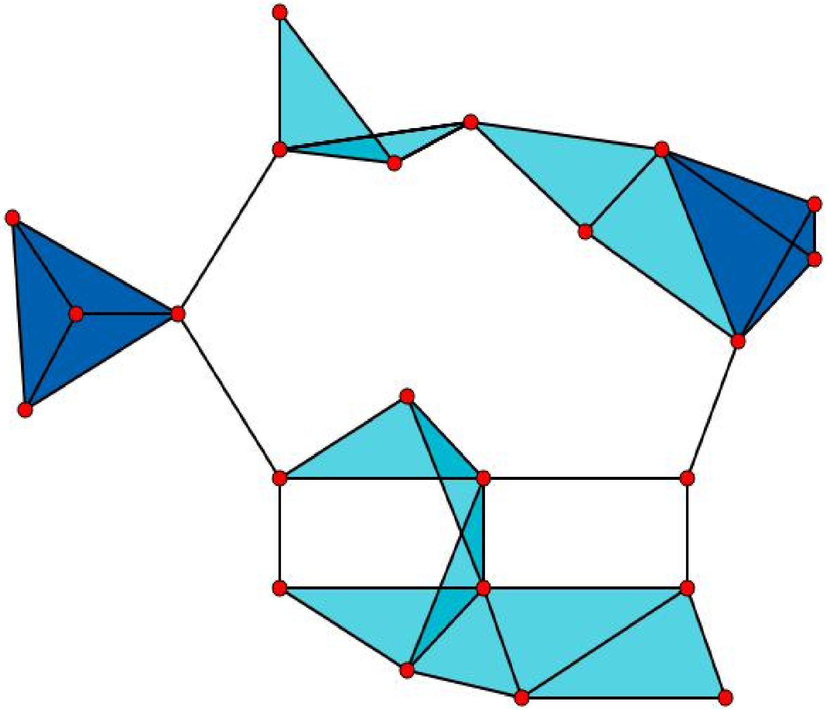
Maximum Clique: the largest *maximal clique* in the graph

Does this graph contain a 4-clique?

Indeed it does!



The CLIQUE Problem



A graph with

- 23 × 1-vertex cliques (the vertices),
- 42 × 2-vertex cliques (the edges),
- 19 × 3-vertex cliques (light and dark blue triangles), and
- 2 × 4-vertex cliques (dark blue areas).

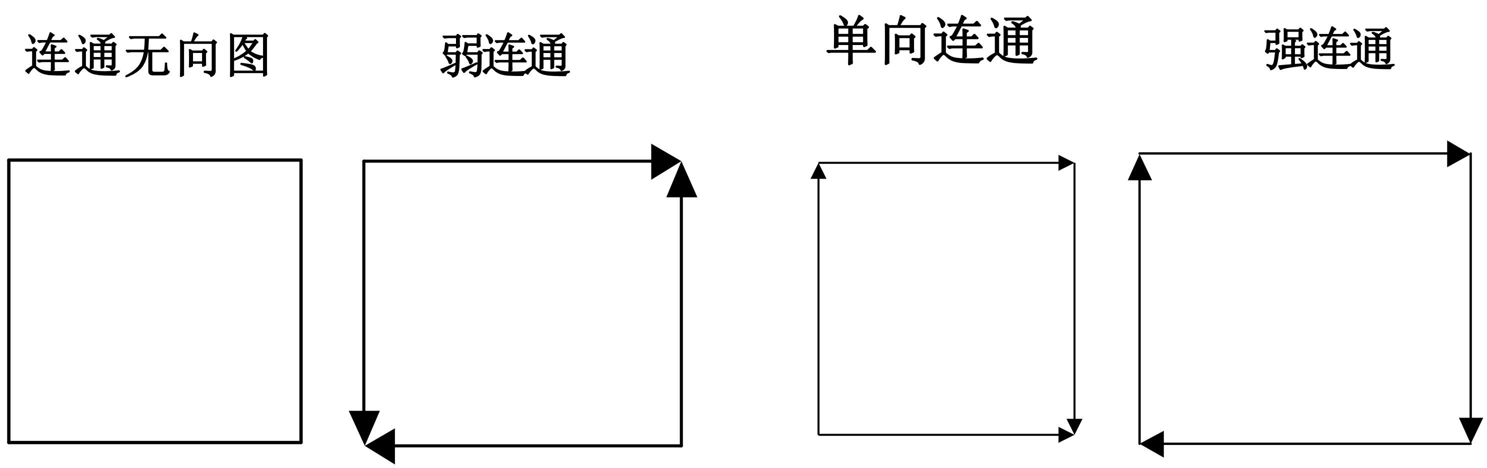
The 11 light blue triangles form maximal cliques. The two dark blue 4-cliques are both maximum and maximal, and the clique number of the graph is 4.

——From Wikipedia

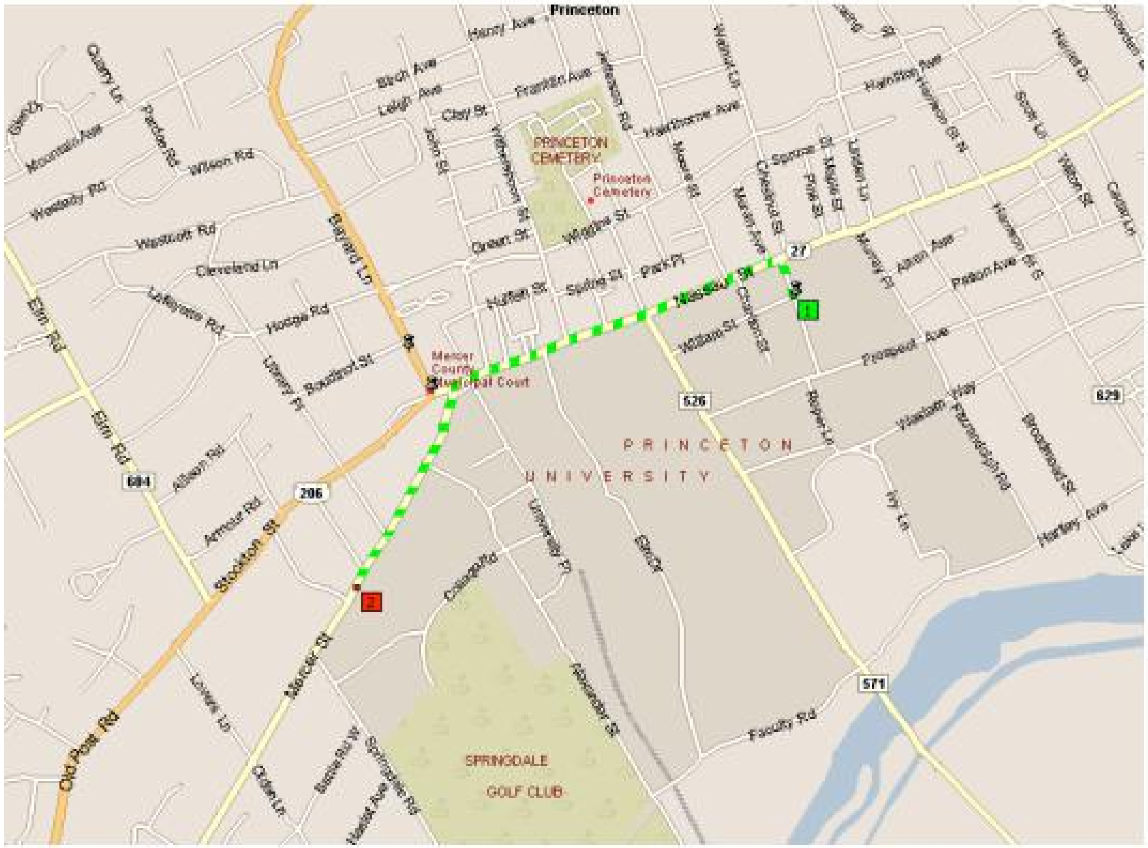
有向连通图

弱连通、单向连通、强连通图

示例 12

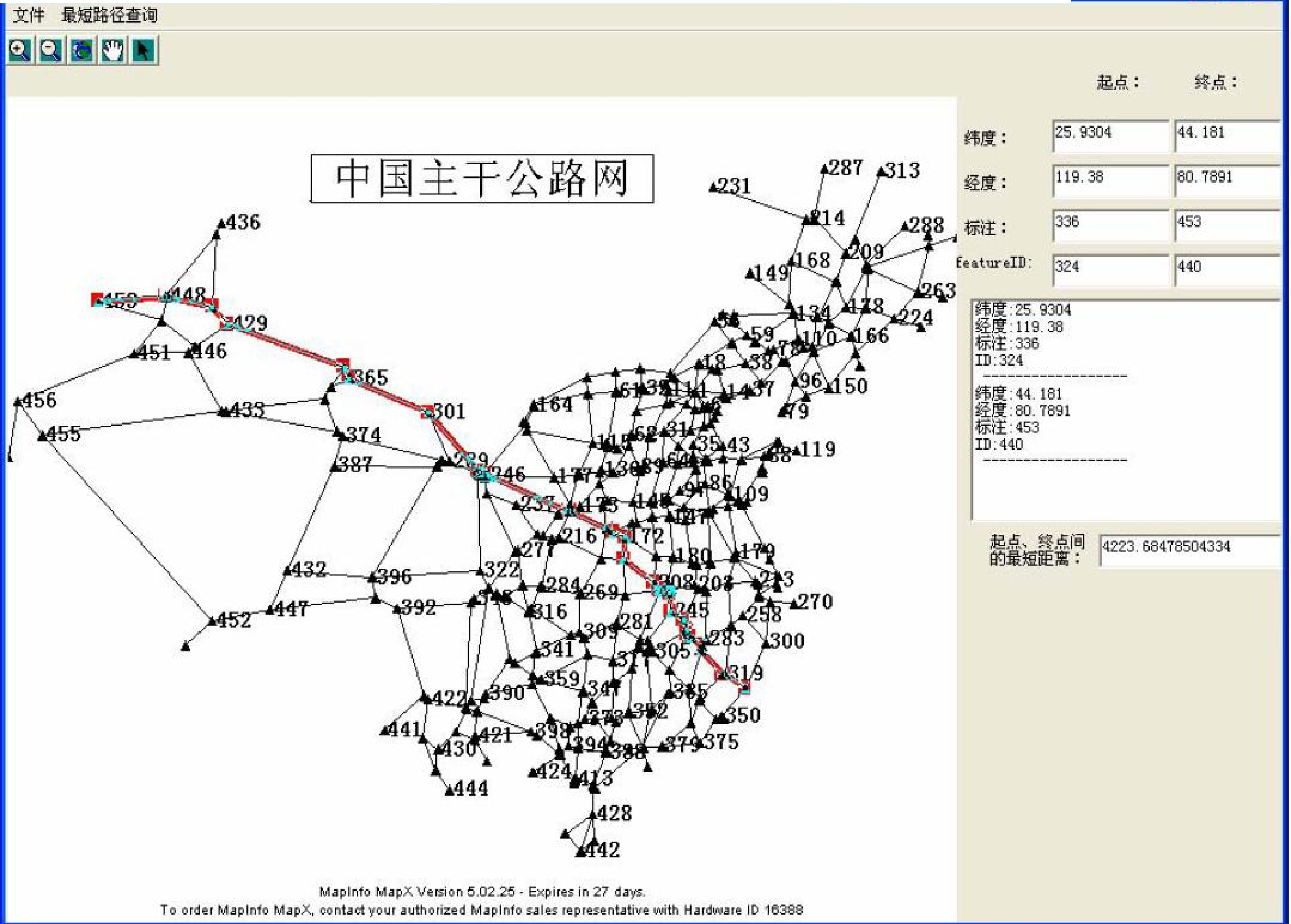


讨论1 Dijkstra算法



shortest path from Princeton CS department to Einstein's house

基于Dijkstra算法的公路网最短路径查询实现



讨论1 Dijkstra算法

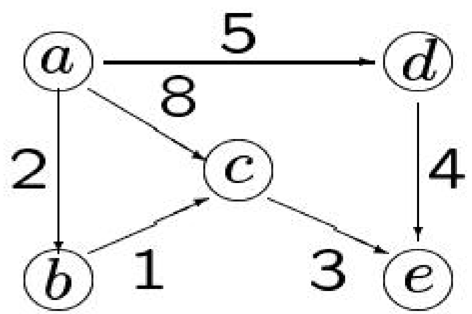
The Problem: Given a digraph with positive edge weights $G = (V, E)$
and a distinguished *source vertex*, $s \in V$,
determine the distance and a shortest path from the source vertex to every vertex in the digraph.

Question: How do you design an efficient algorithm for this problem?

Dijkstra Algorithm

Important Observation: Any subpath of a shortest path must also be a shortest path. Why?

Example: In the following digraph, $\langle a, b, c, e \rangle$ is a shortest path. The subpath $\langle a, b, c \rangle$ is also a shortest path.



length($\langle a, b, c, e \rangle$) $= 6$
distance from a to e is 6

Observation Extending this idea we observe the existence of a *shortest path tree* in which distance from source to vertex v is length of shortest path from source to vertex in original tree.

Intuition behind Dijkstra's Algorithm

Construct the shortest path tree edge by edge; at each step adding one new edge, corresponding to construction of shortest path to the current new vertex.

Edge relaxation

For all v, dist[v] is the length of some path from s to v.

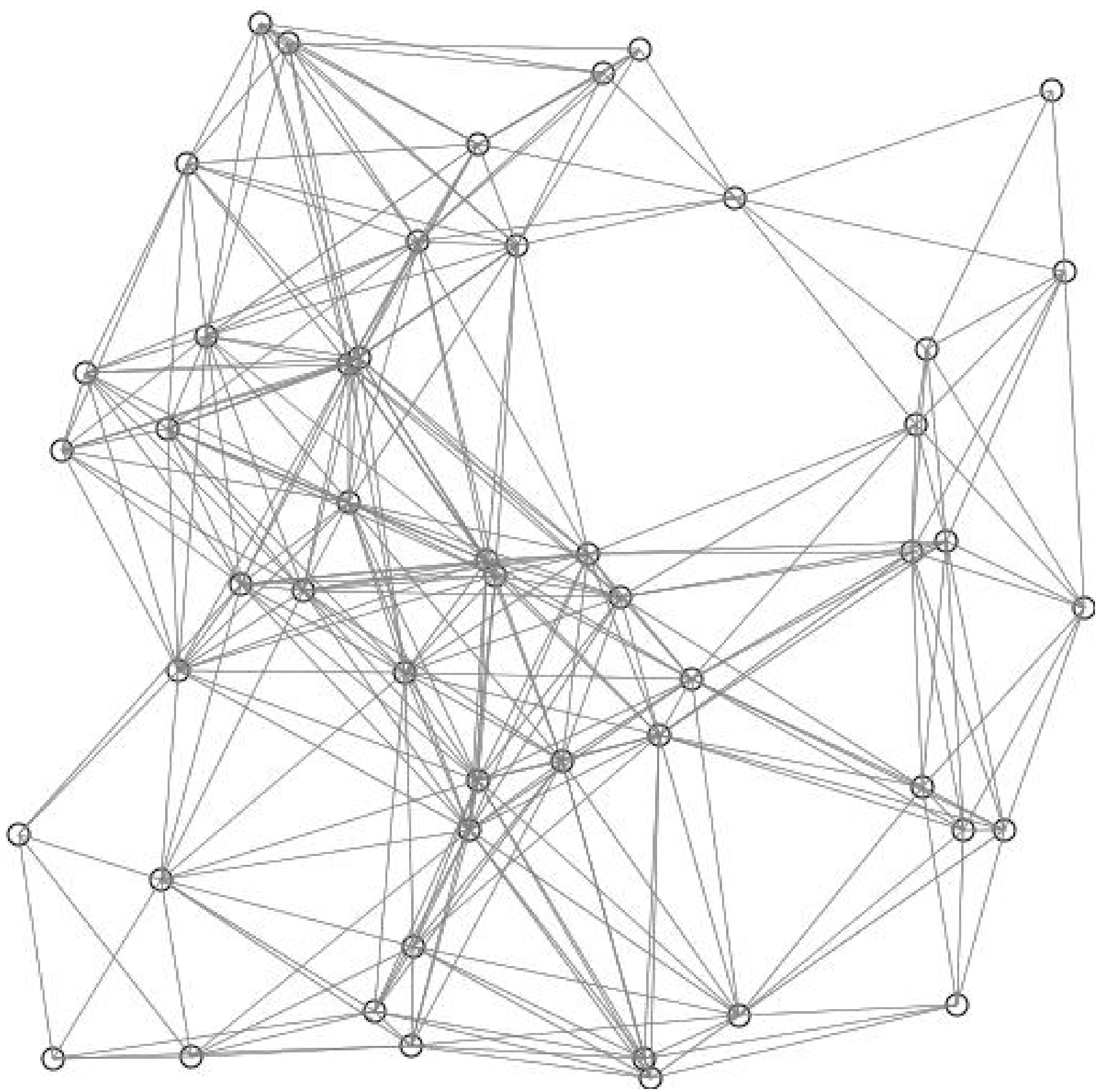
Relaxation along edge e from v to w.

- dist[v] is length of some path from s to v
- dist[w] is length of some path from s to w
- if v-w gives a shorter path to w through v, update dist[w] and pred[w]

```
if (dist[w] > dist[v] + e.weight())
{
    dist[w] = dist[v] + e.weight());
    pred[w] = e;
}
```

Relaxation sets dist[w] to the length of a shorter path from s to w (if v-w gives one)

A demo of Dijkstra's algorithm



Greedy

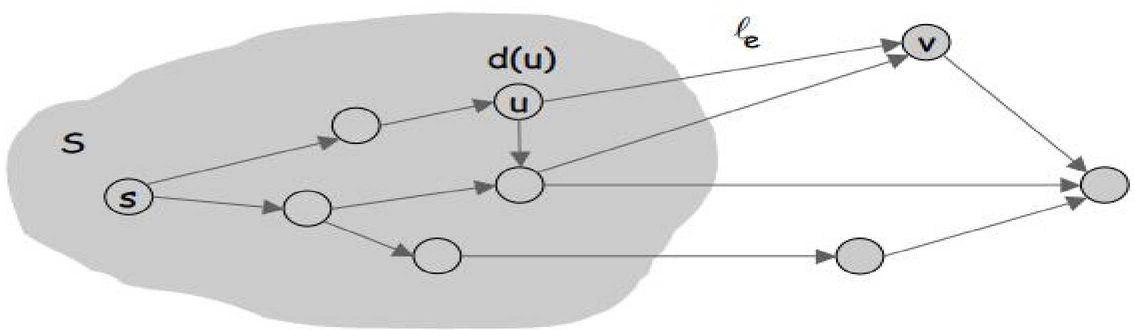
Dijkstra's algorithm.

- Maintain a set of explored nodes S for which we have determined the shortest path distance d(u) from s to u.
- Initialize S = { s }, d(s) = 0.
- Repeatedly choose unexplored node v which minimizes

$$\pi(v) = \min_{e = (u,v) : u \in S} d(u) + \ell_e ,$$

add v to S, and set d(v) = π(v).

shortest path to some u in explored part, followed by a single edge (u, v)



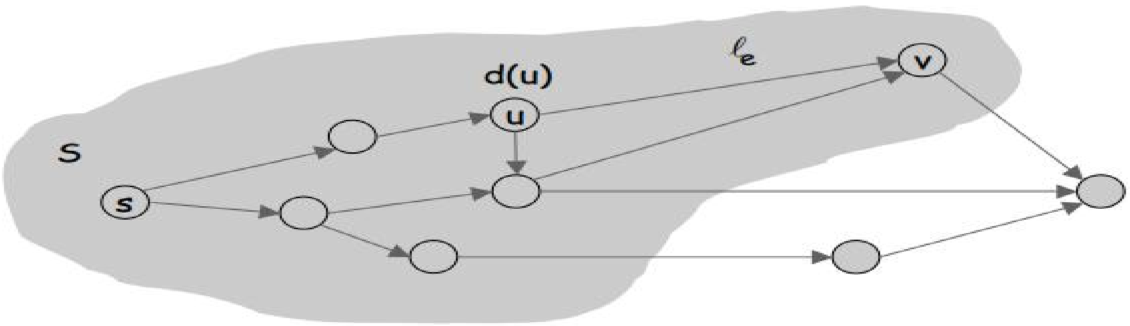
Dijkstra's algorithm.

- Maintain a set of explored nodes S for which we have determined the shortest path distance d(u) from s to u.
- Initialize S = { s }, d(s) = 0.
- Repeatedly choose unexplored node v which minimizes

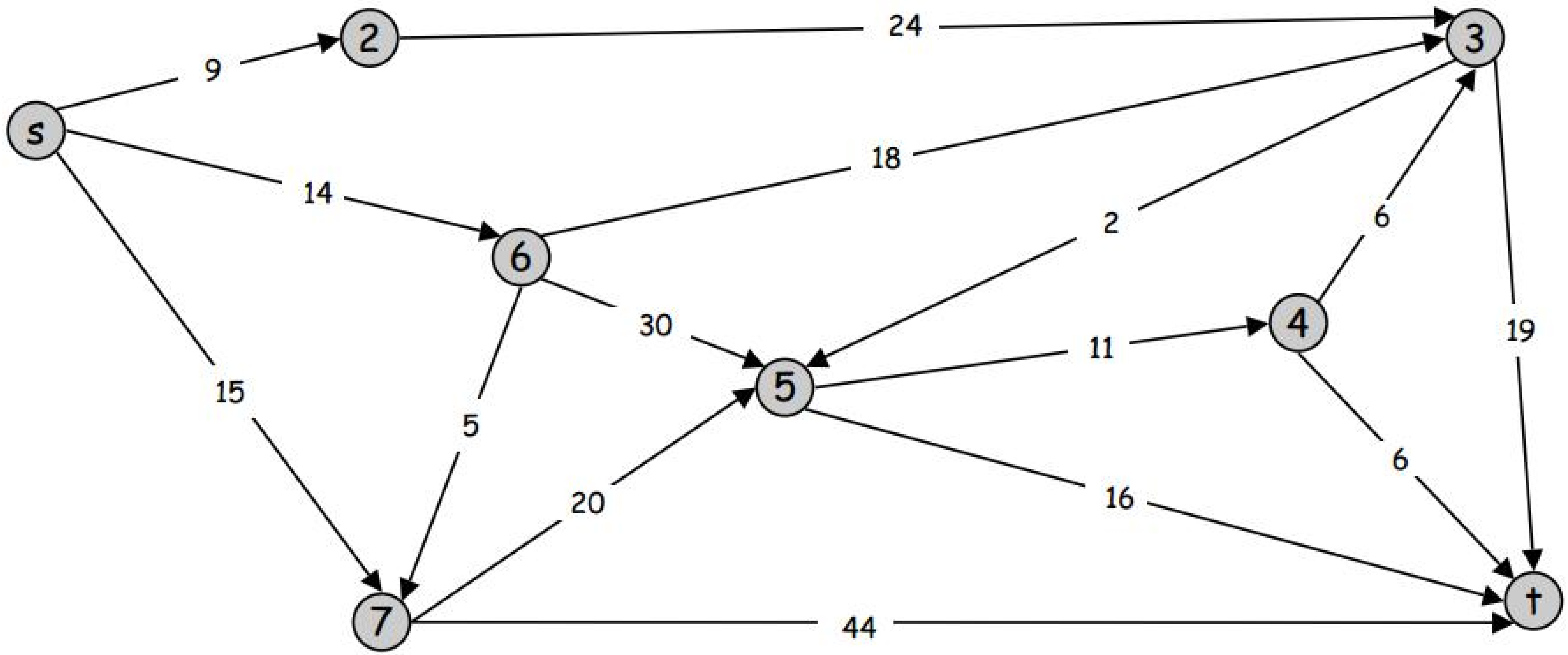
$$\pi(v) = \min_{e = (u,v) \,:\, u \in S} d(u) + \ell_e ,$$

add v to S, and set d(v) = π(v).

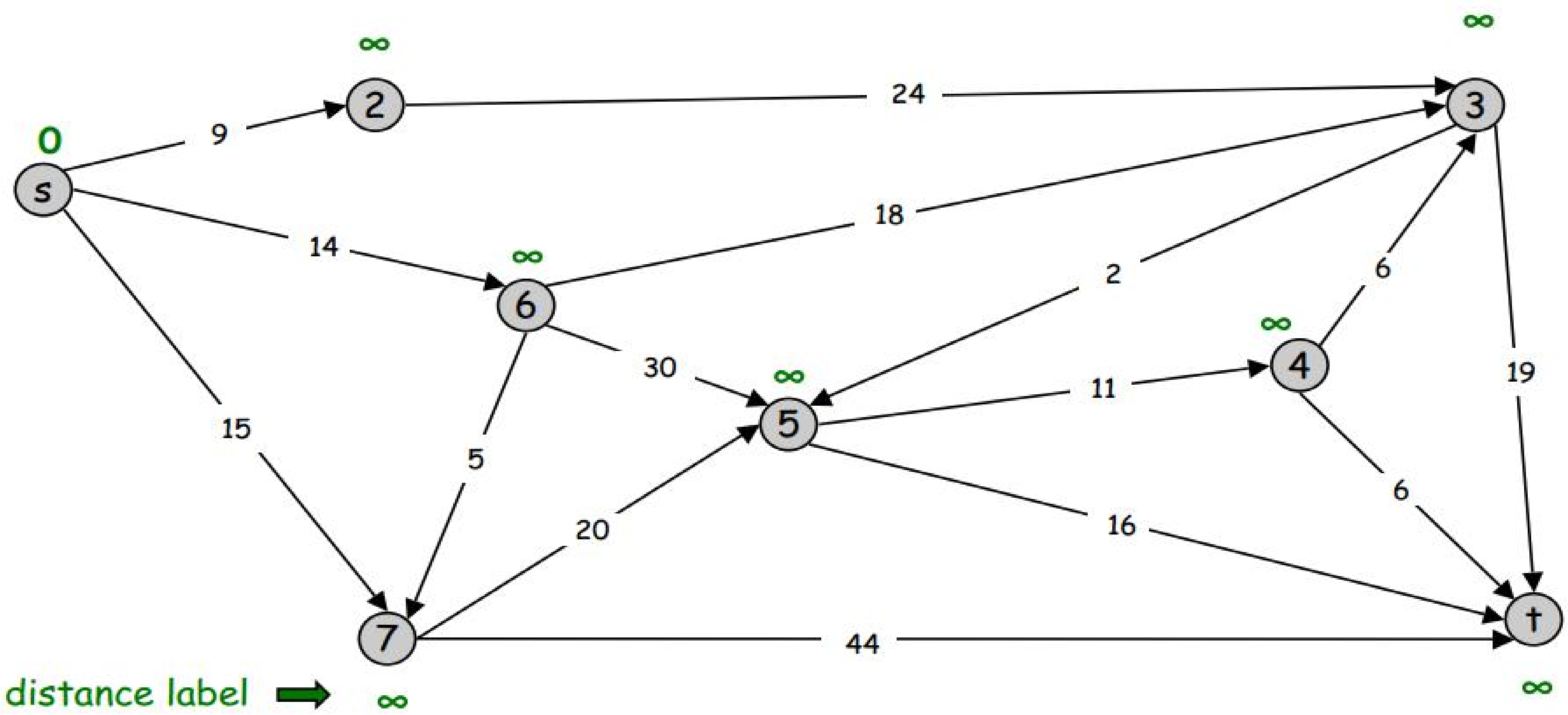
shortest path to some u in explored part, followed by a single edge (u, v)



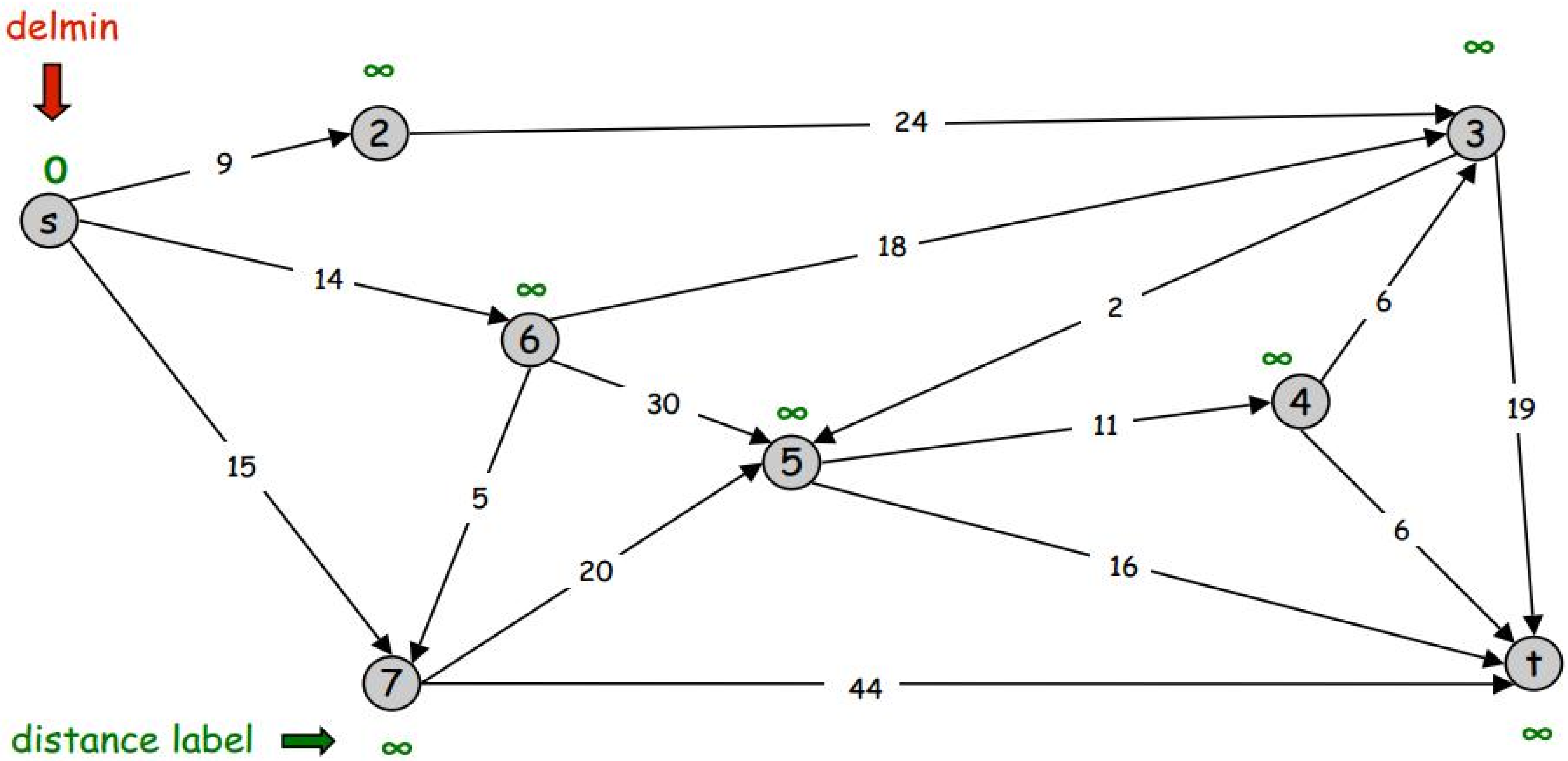
An illustration: find shortest directed path from s to t.



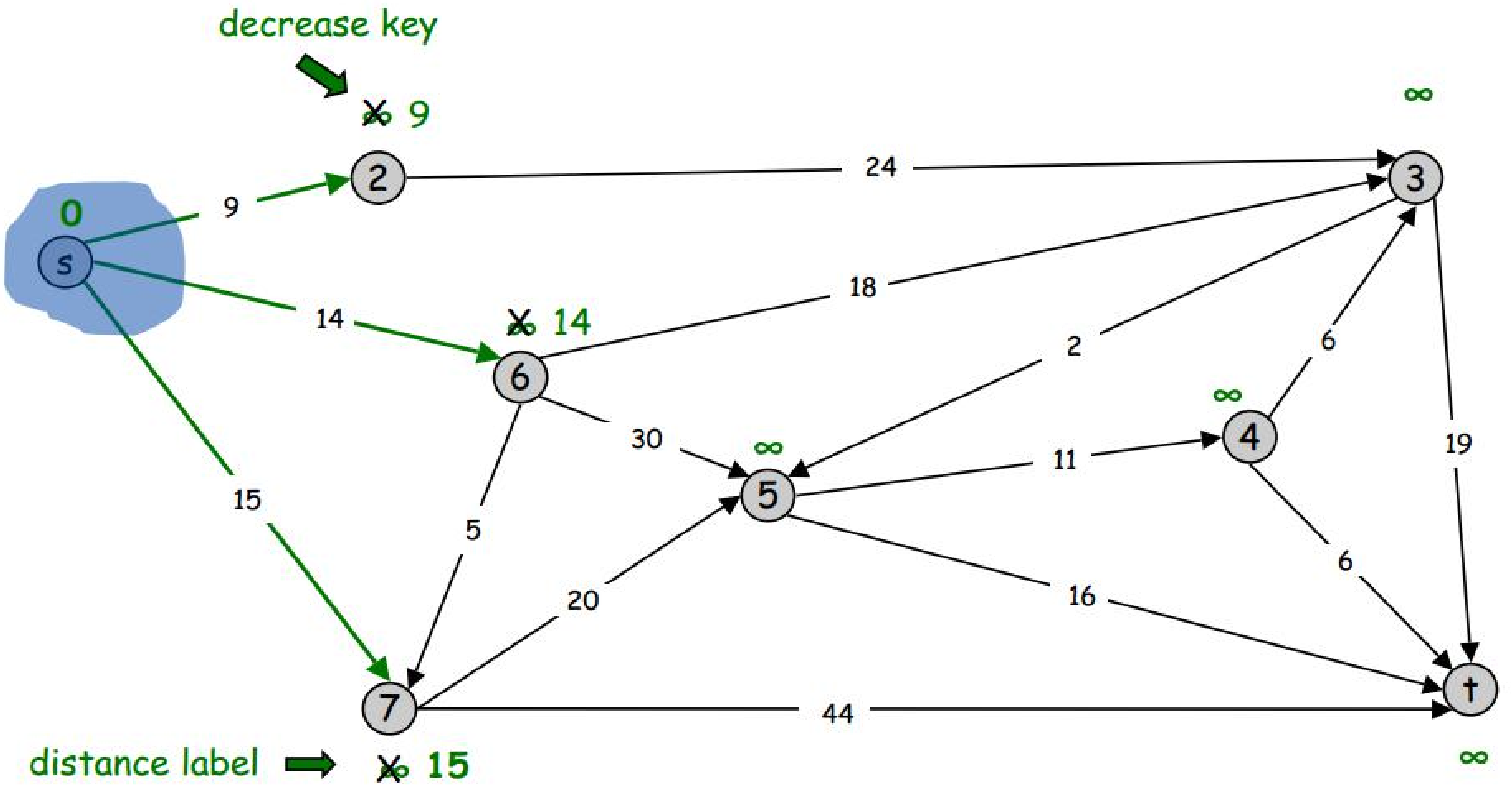
$S = \{\ \}$
$PQ = \{ s, 2, 3, 4, 5, 6, 7, t \}$



S = { }
PQ = { s, 2, 3, 4, 5, 6, 7, t }

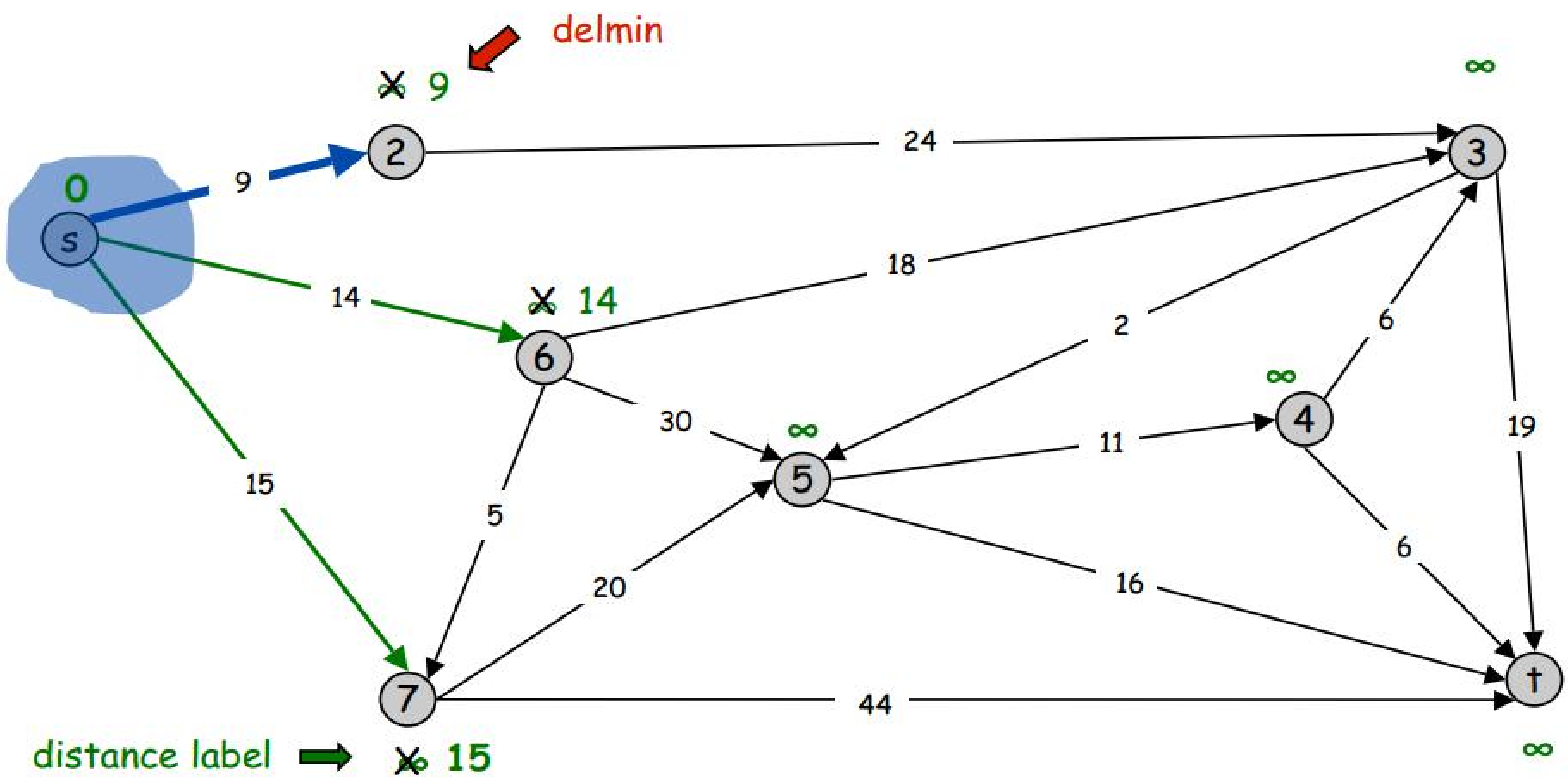


$S = \{ s \}$
$PQ = \{ 2, 3, 4, 5, 6, 7, t \}$

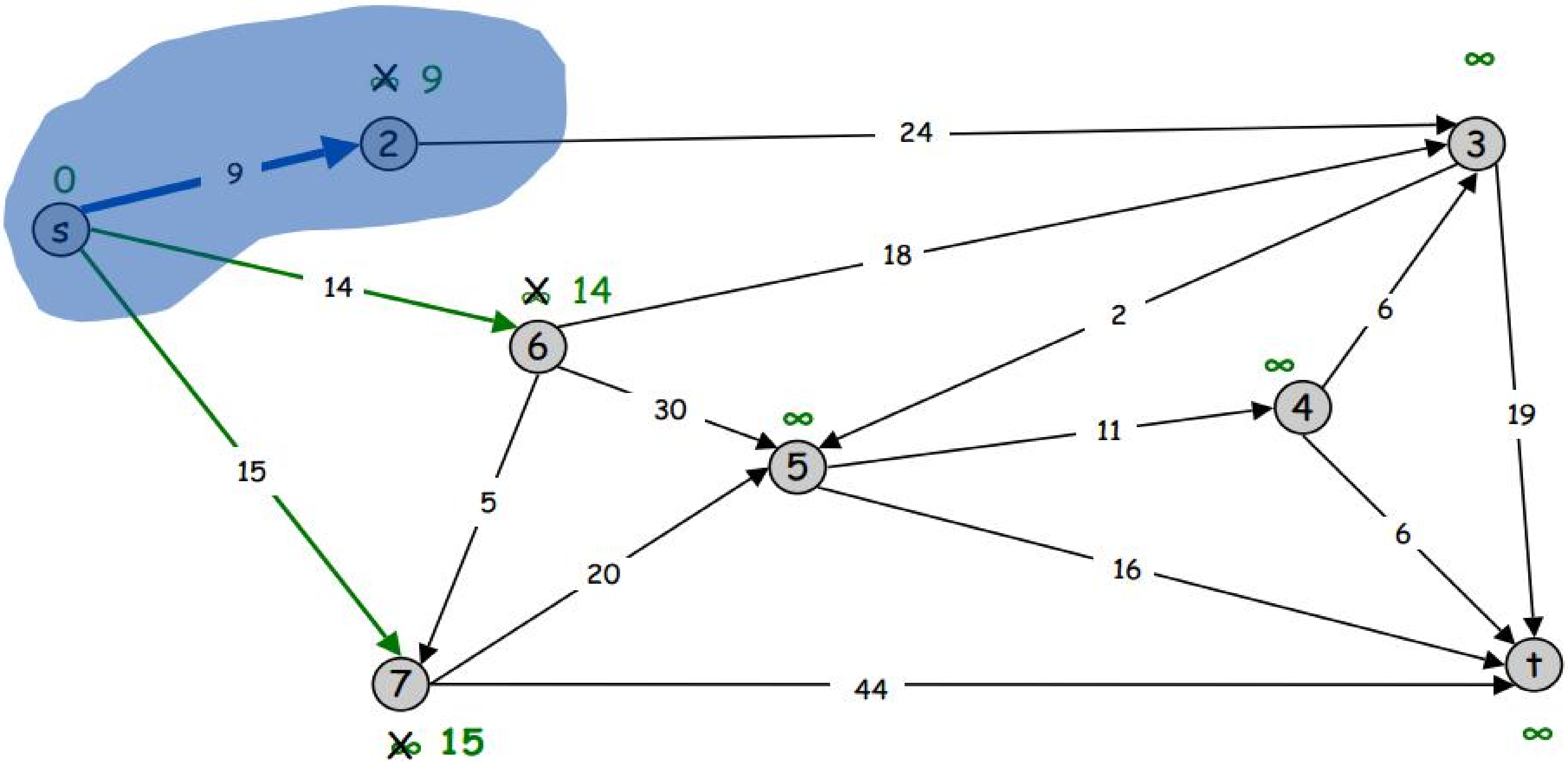


$S = \{ s \}$

$PQ = \{ 2, 3, 4, 5, 6, 7, t \}$

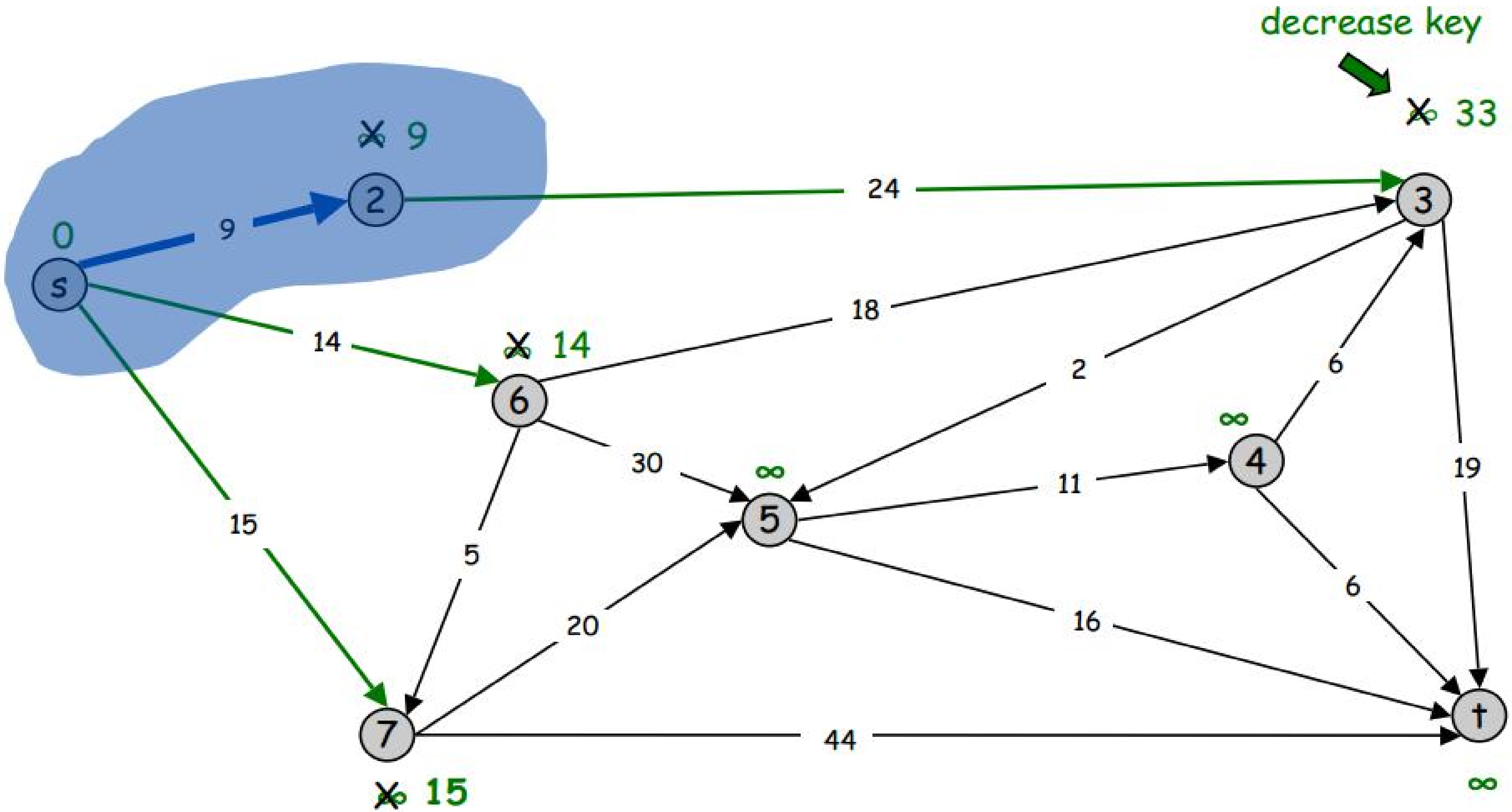


S = { s, 2 }
PQ = { 3, 4, 5, 6, 7, t }

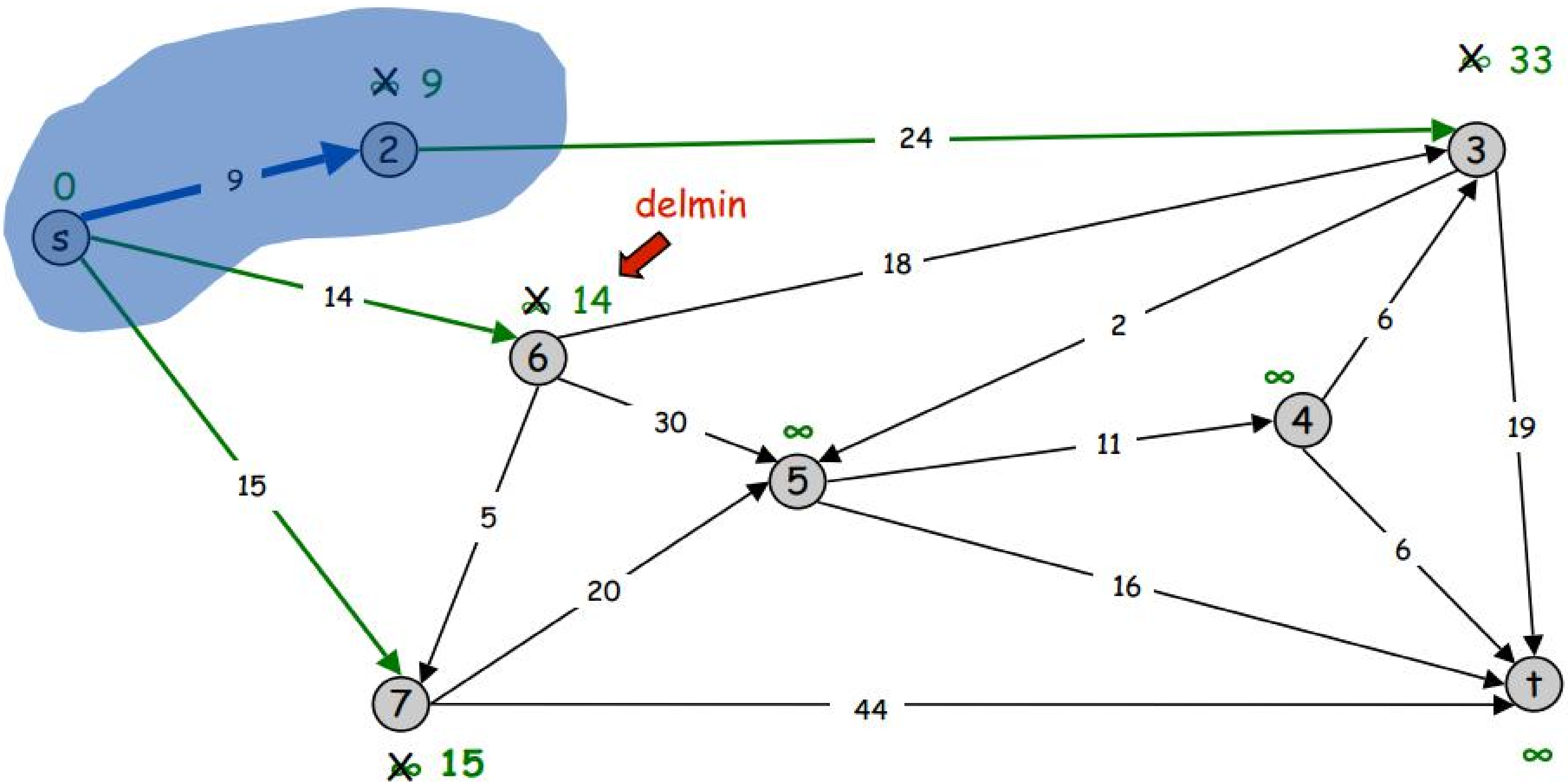


$S = \{ s, 2 \}$

$PQ = \{ 3, 4, 5, 6, 7, t \}$

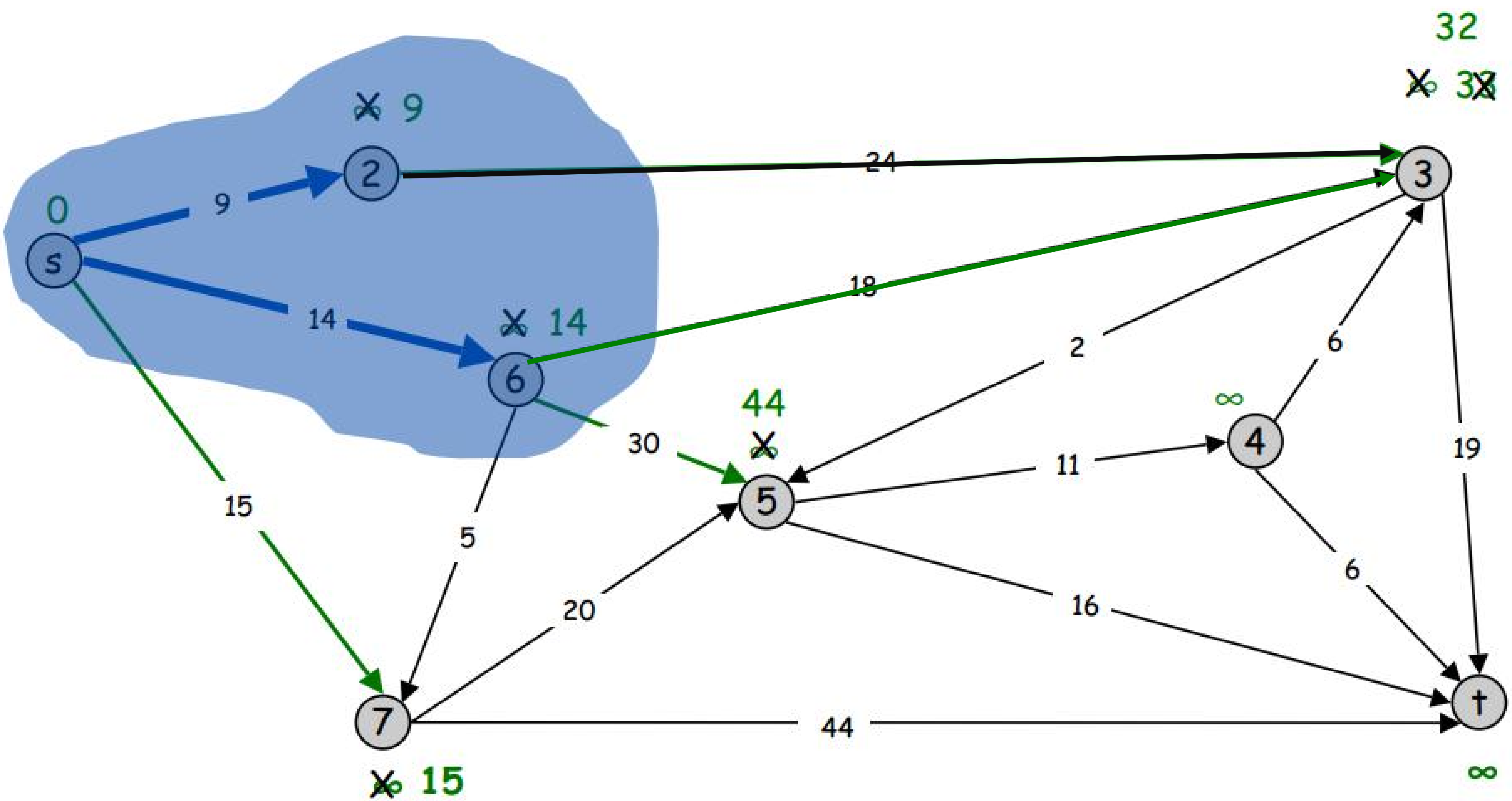


$S = \{ s, 2 \}$
$PQ = \{ 3, 4, 5, 6, 7, t \}$

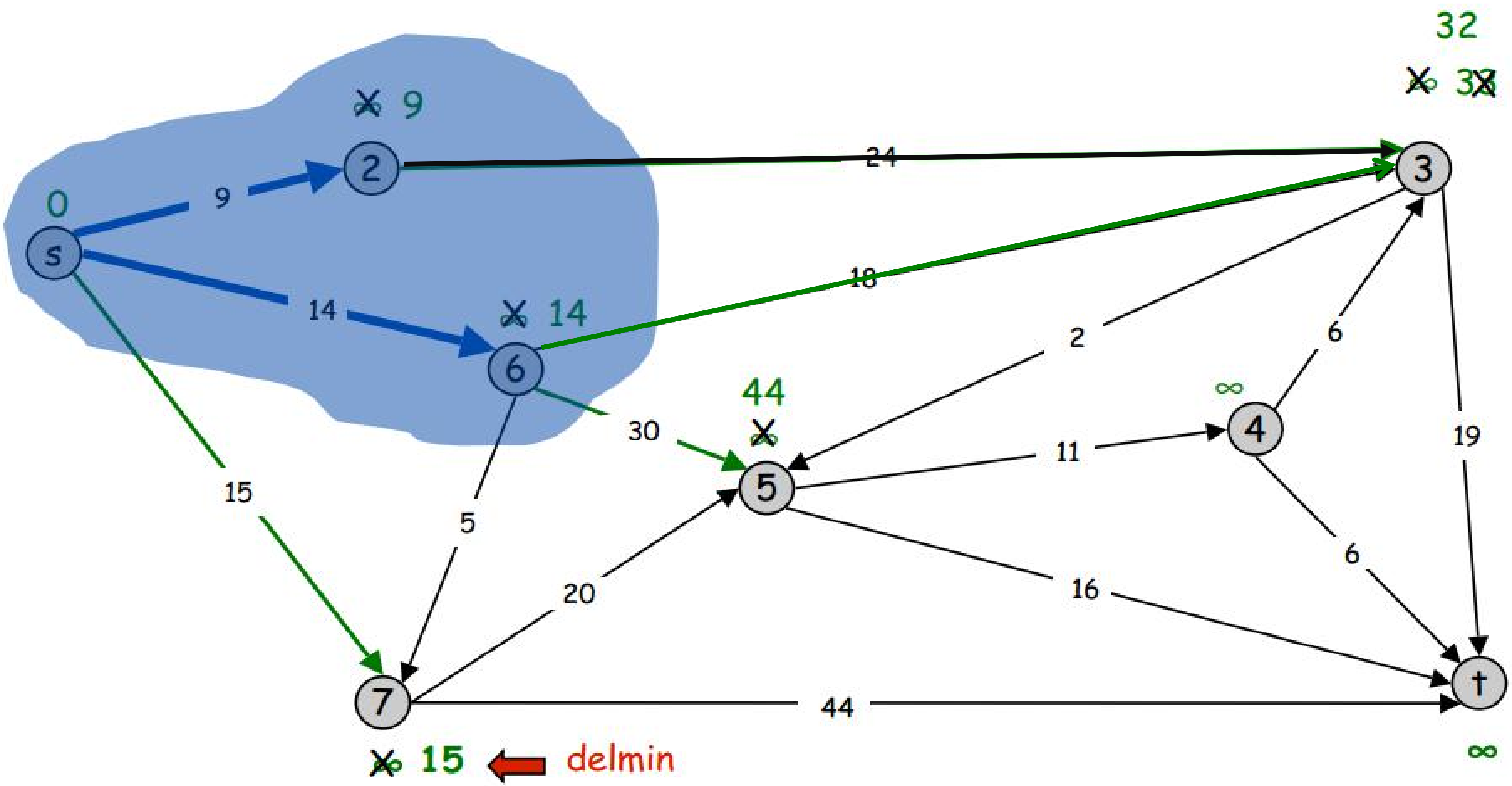


S = { s, 2, 6 }

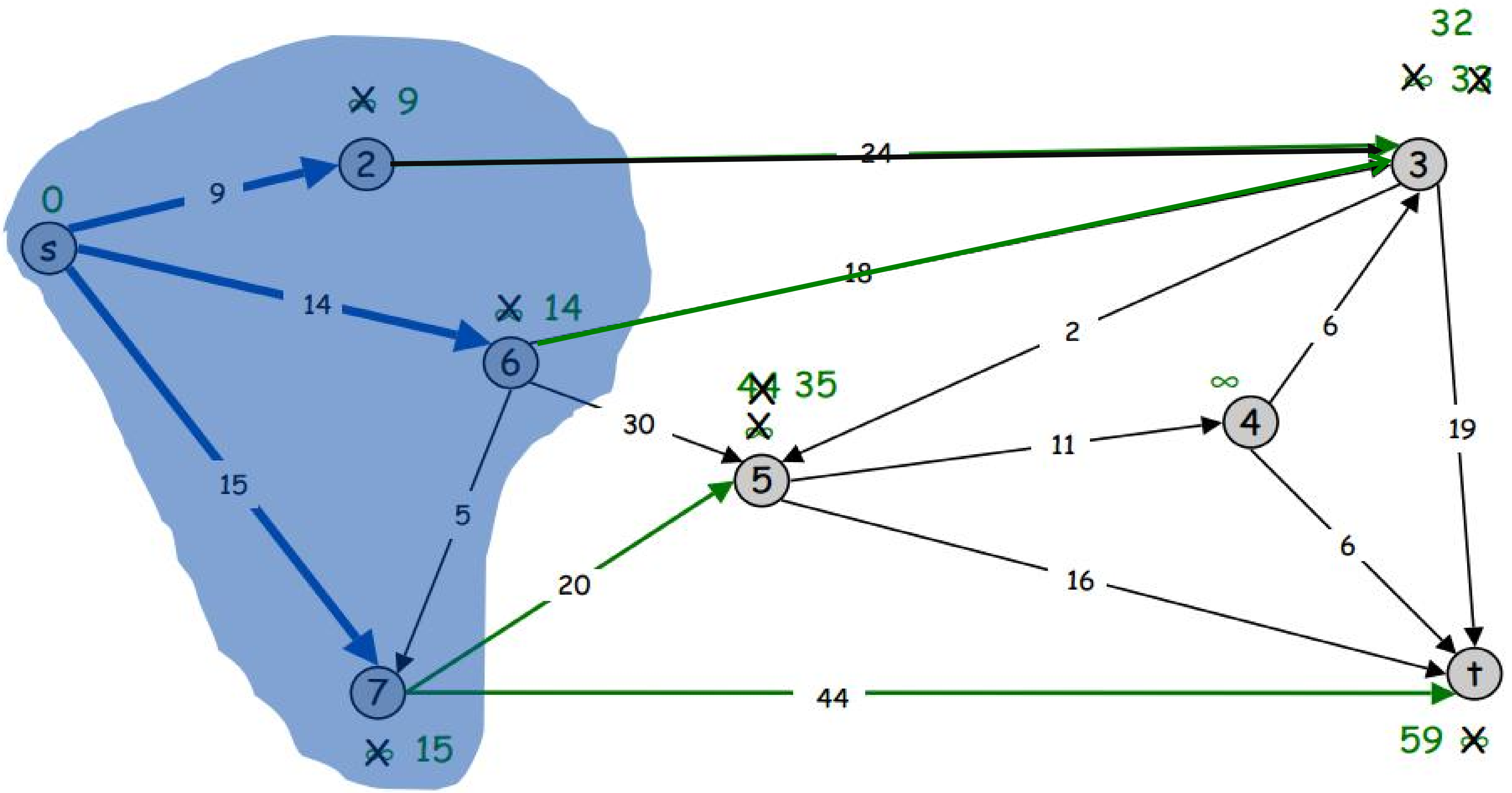
PQ = { 3, 4, 5, 7, t }



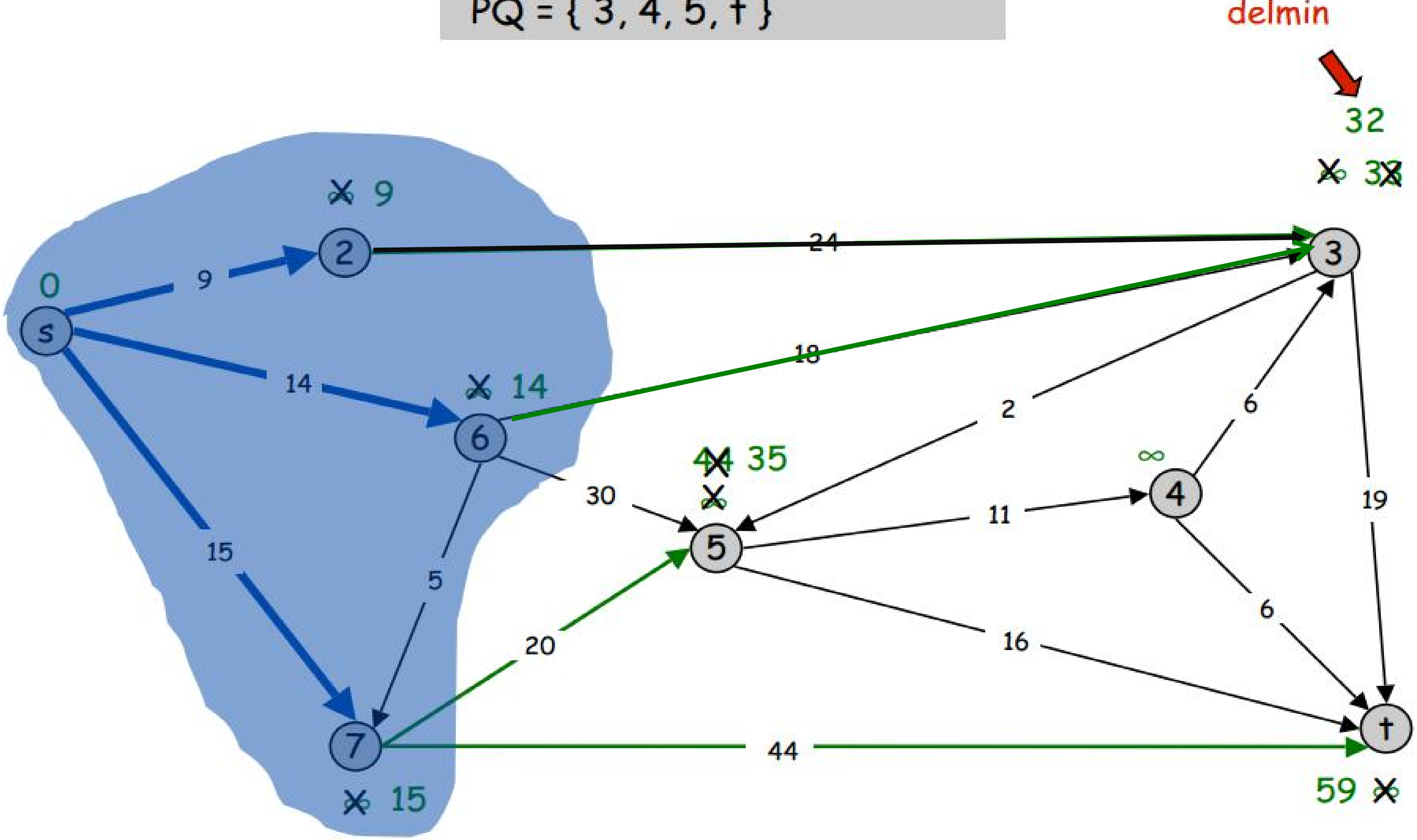
$S = \{ s, 2, 6 \}$
$PQ = \{ 3, 4, 5, 7, t \}$



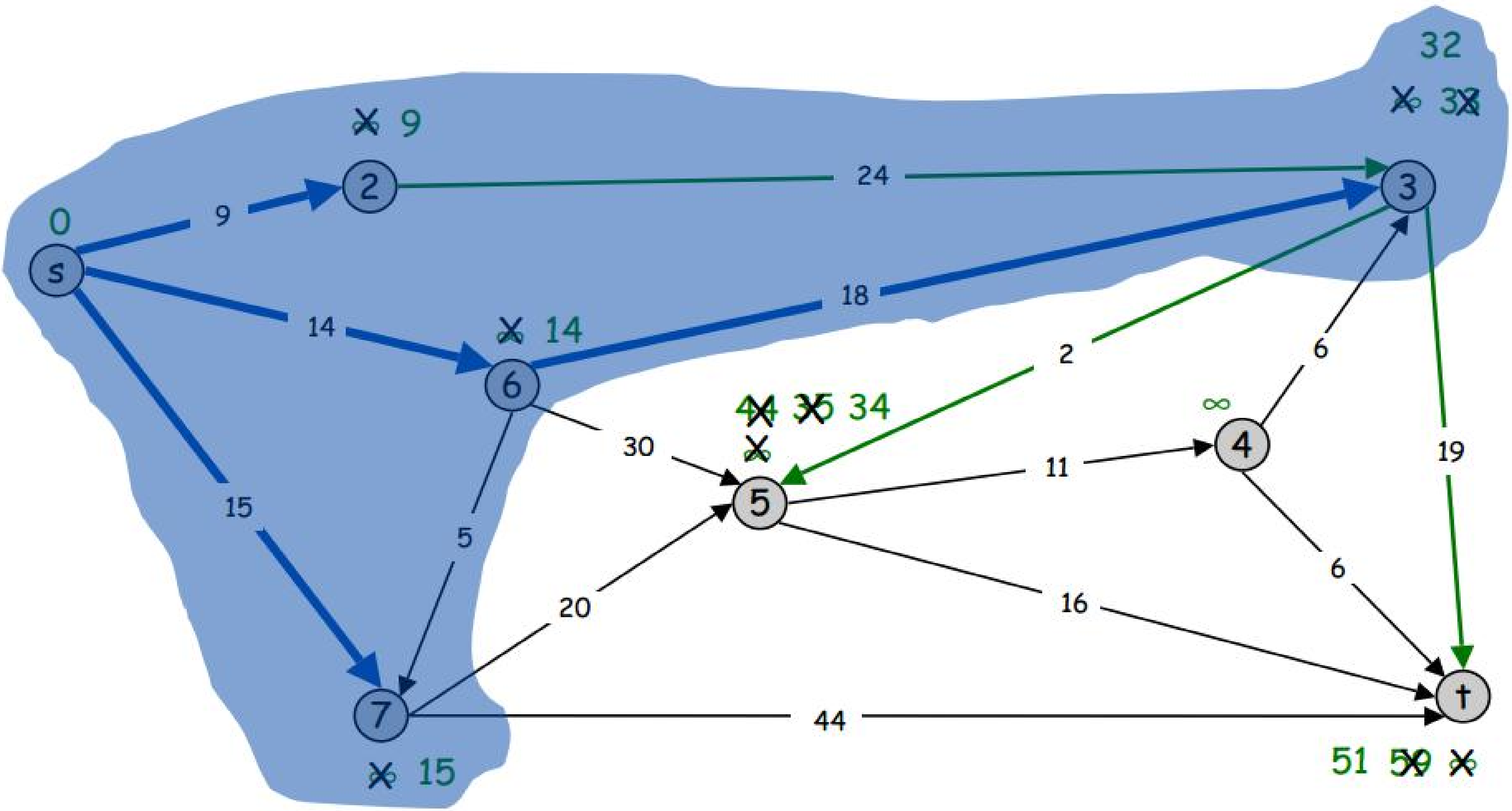
S = { s, 2, 6, 7 }
PQ = { 3, 4, 5, t }



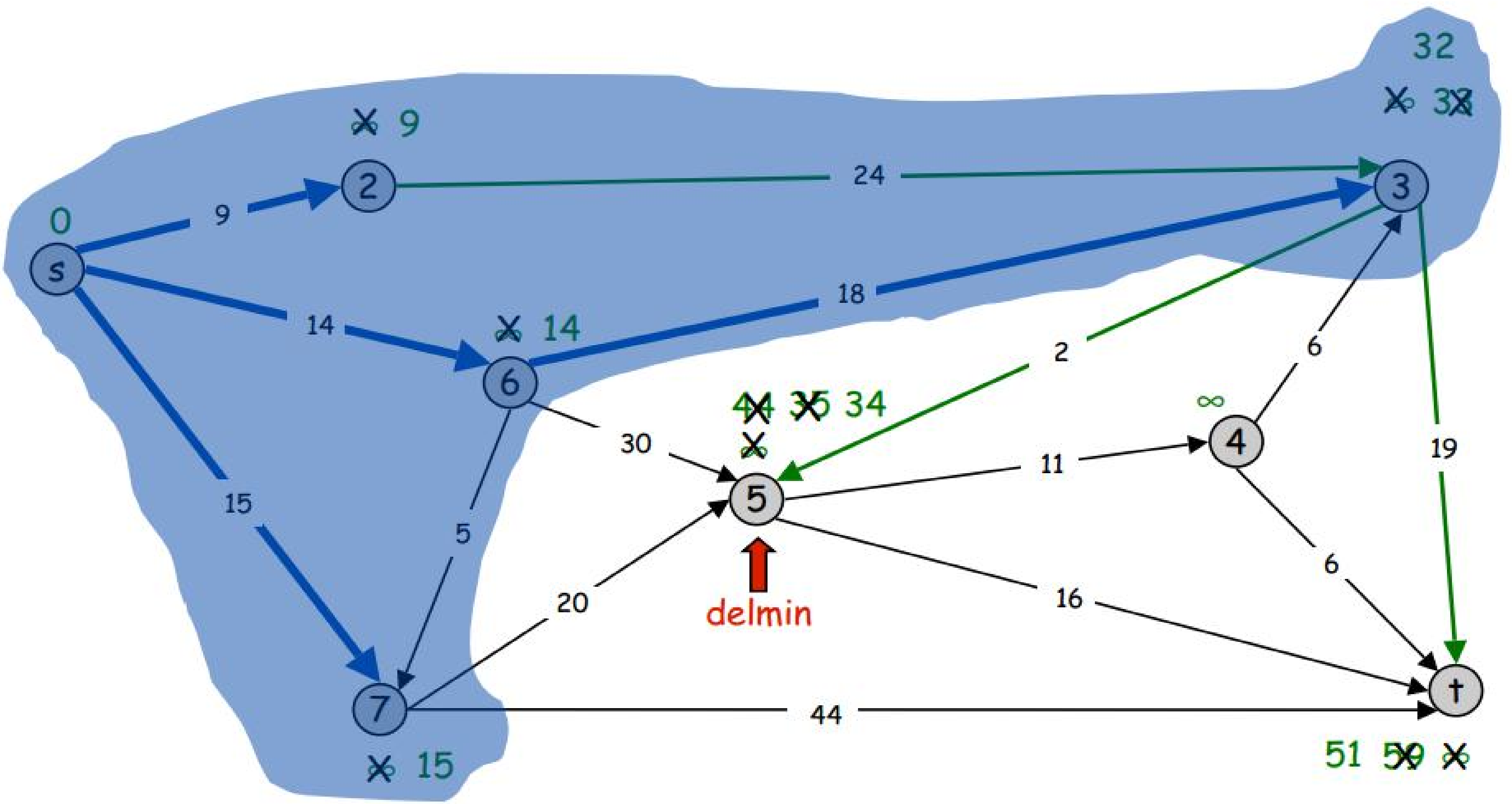
$S = \{ s, 2, 6, 7 \}$
$PQ = \{ 3, 4, 5, t \}$



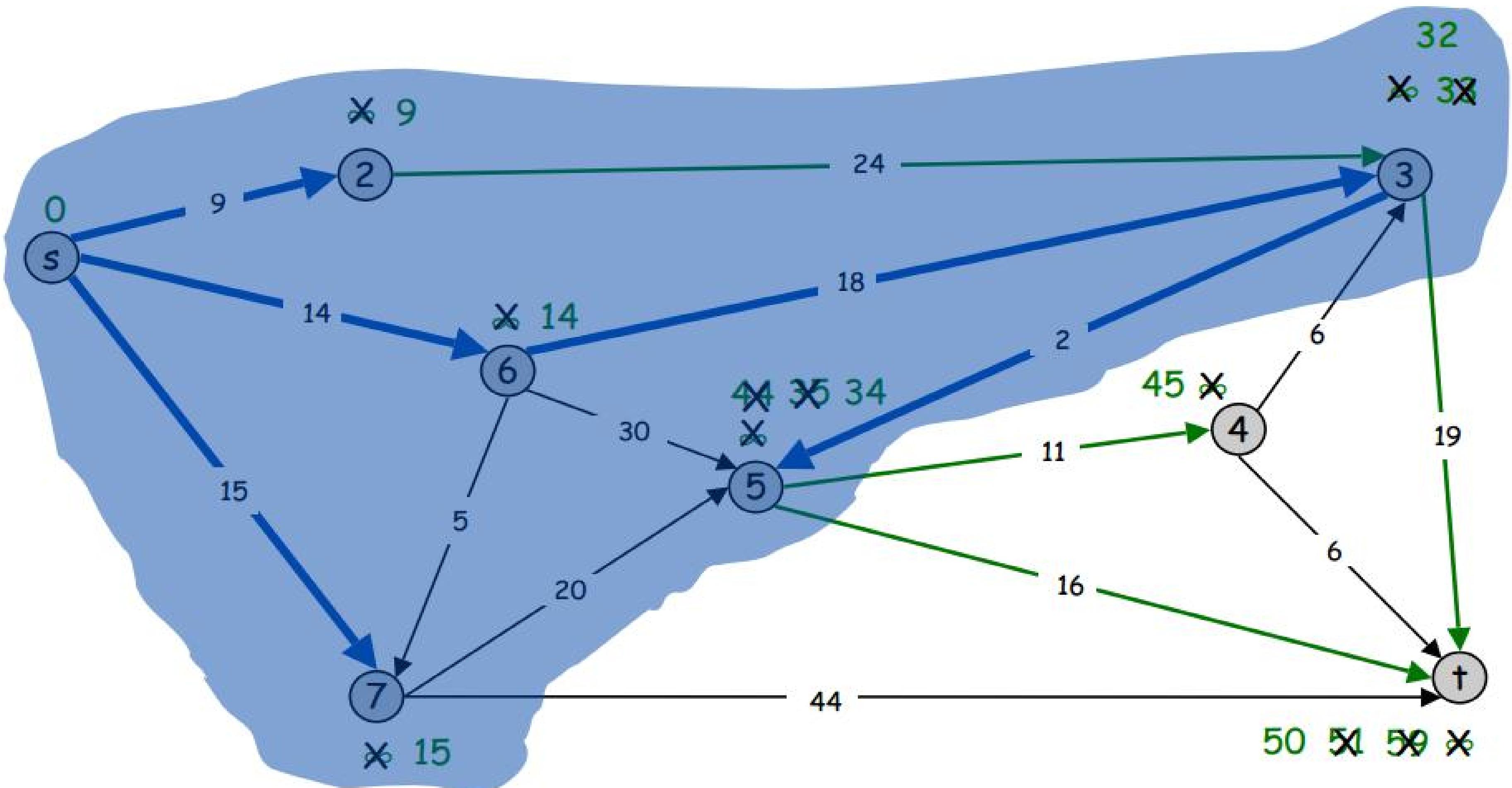
S = { s, 2, 3, 6, 7 }
PQ = { 4, 5, t }



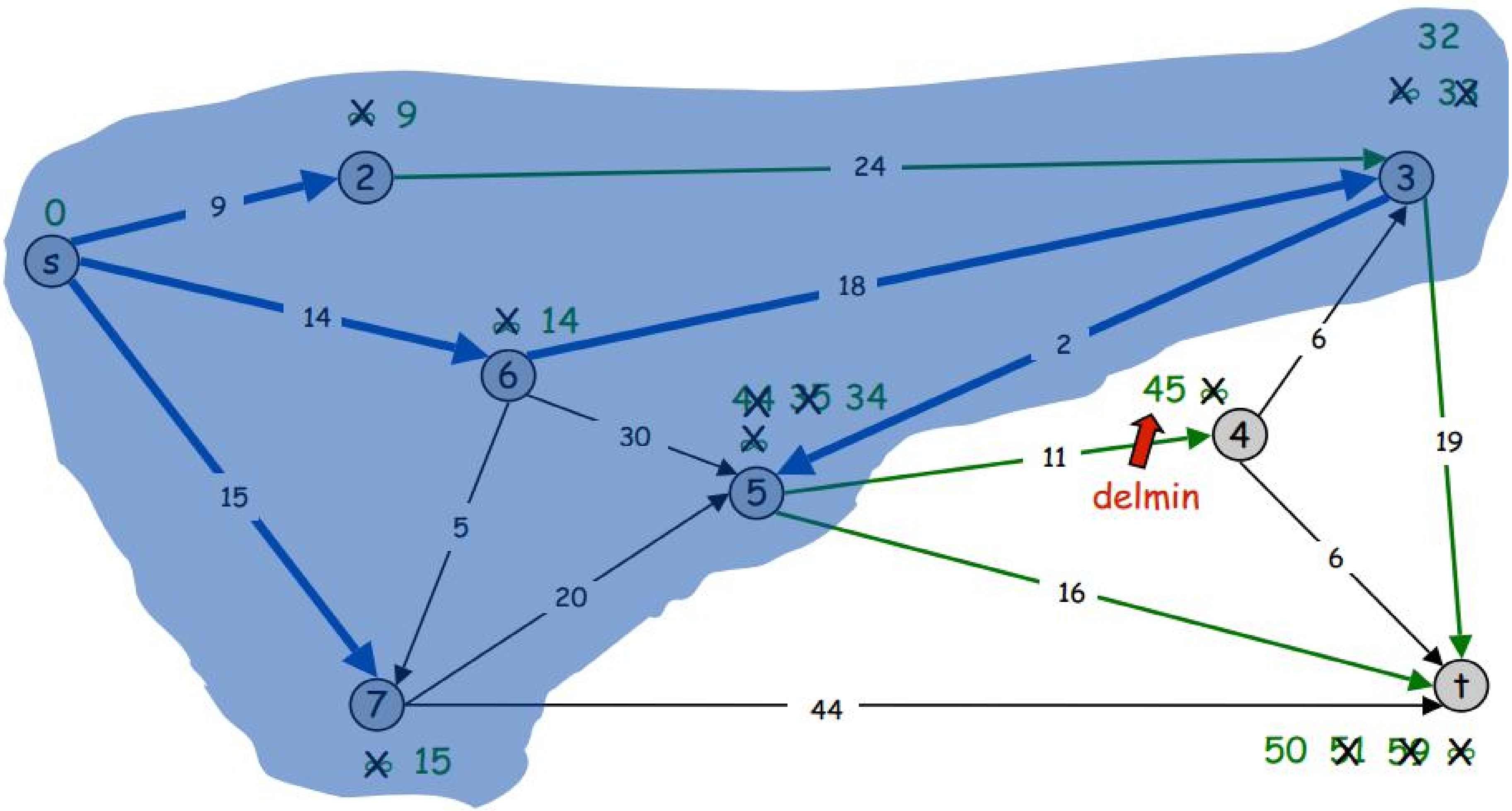
S = { s, 2, 3, 6, 7 }
PQ = { 4, 5, t }



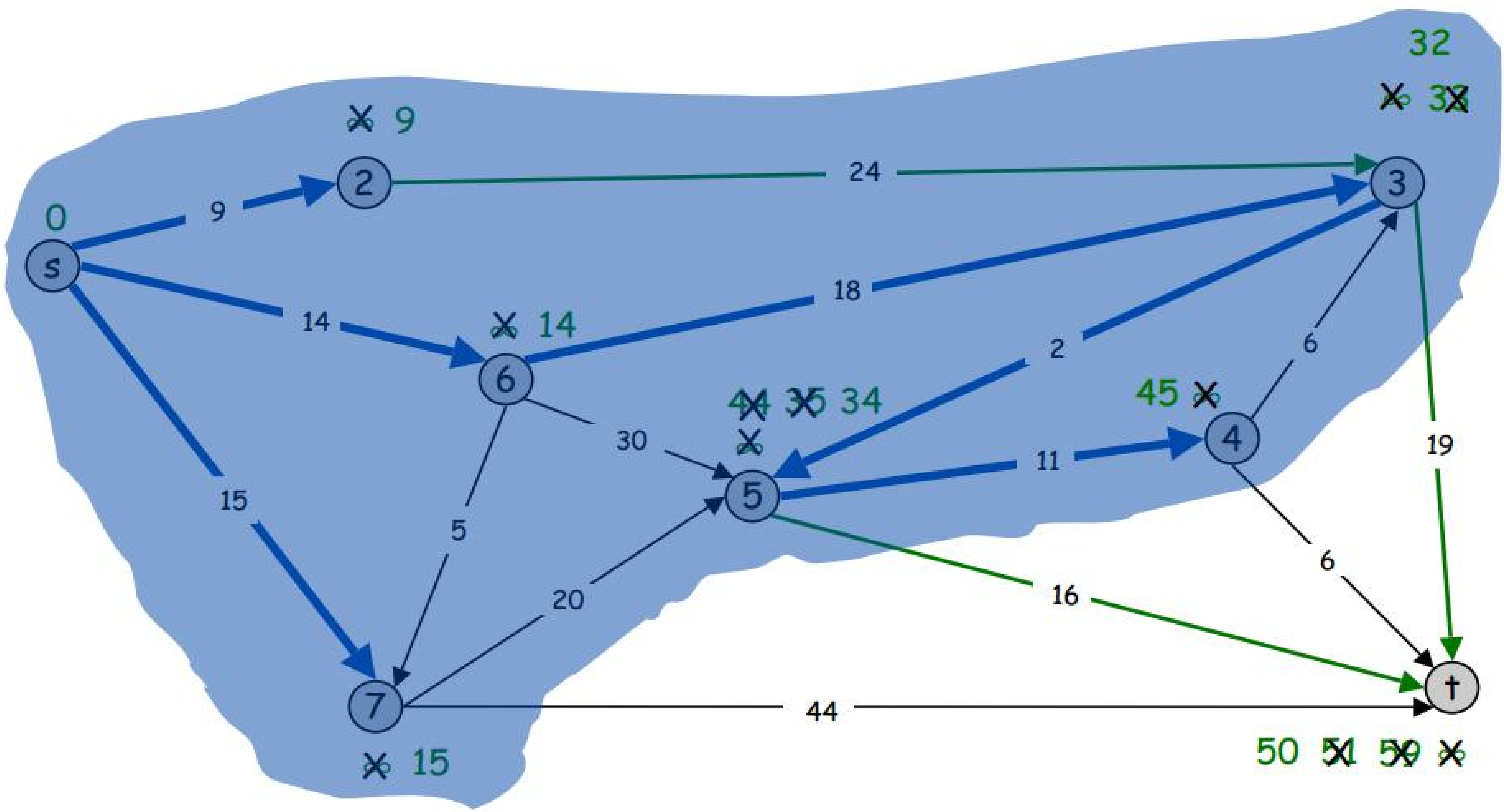
S = { s, 2, 3, 5, 6, 7 }
PQ = { 4, t }



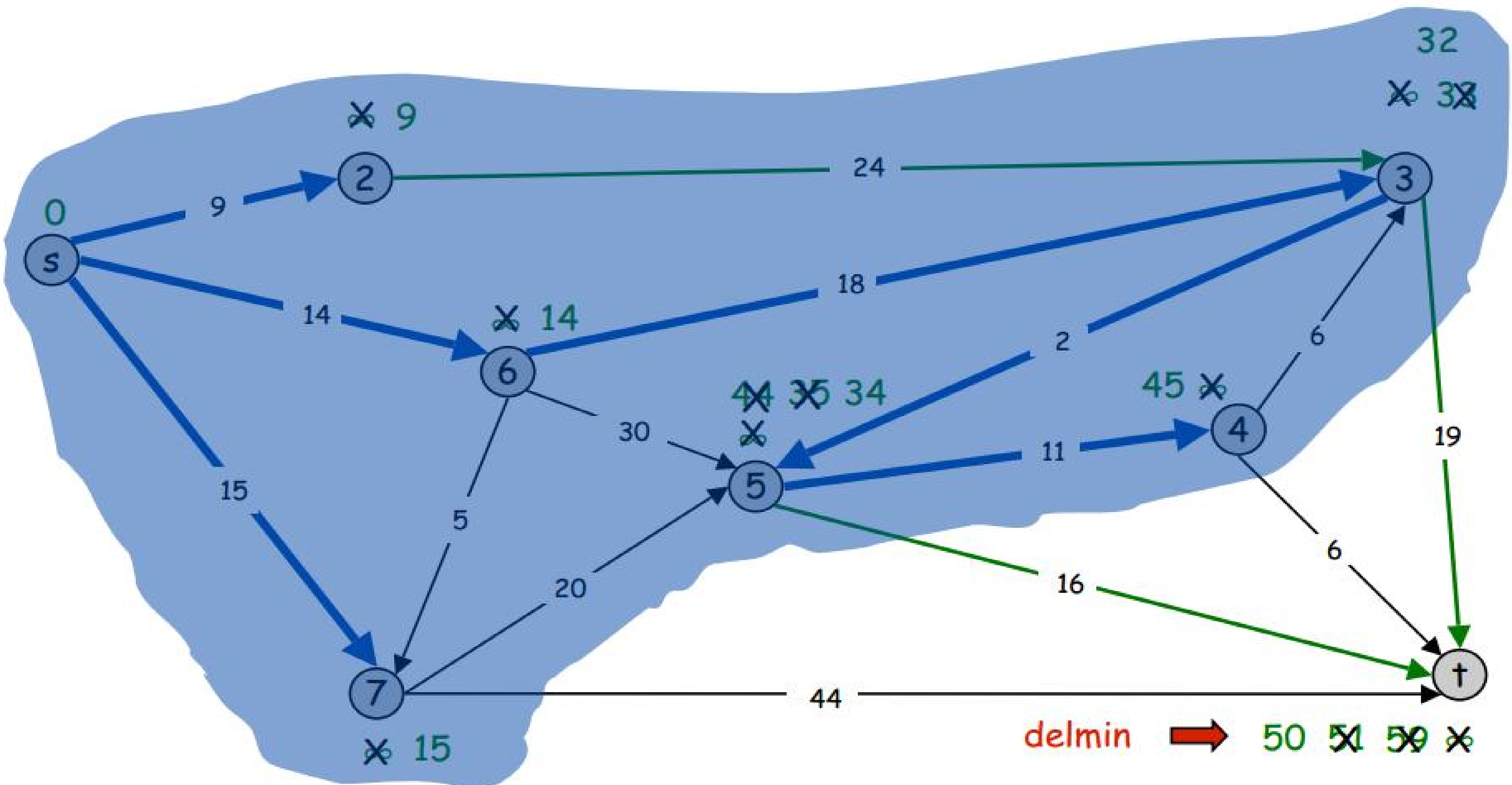
S = { s, 2, 3, 5, 6, 7 }
PQ = { 4, t }



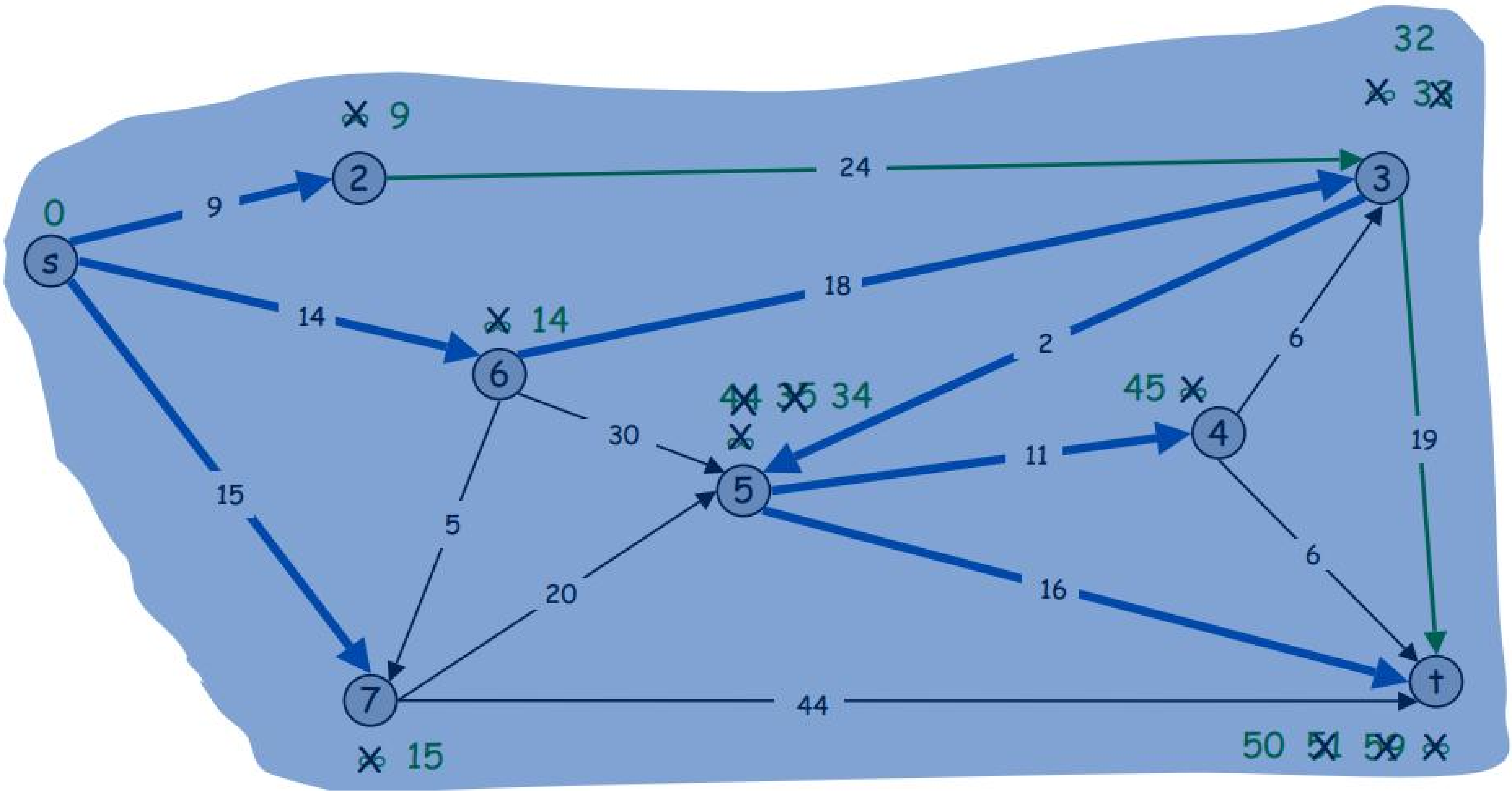
S = { s, 2, 3, 4, 5, 6, 7 }
PQ = { t }



S = { s, 2, 3, 4, 5, 6, 7 }
PQ = { t }

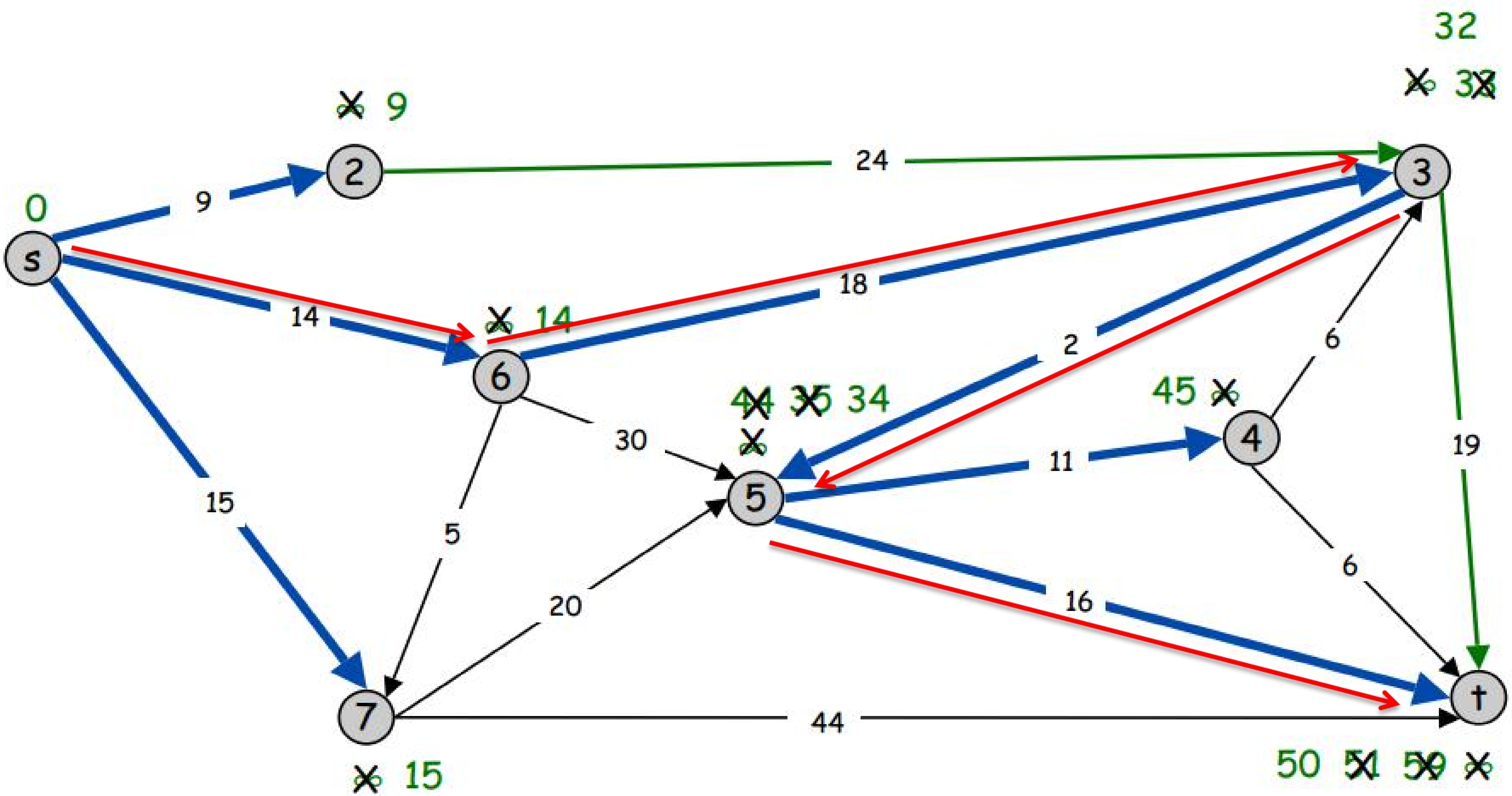


S = { s, 2, 3, 4, 5, 6, 7, t }
PQ = { }



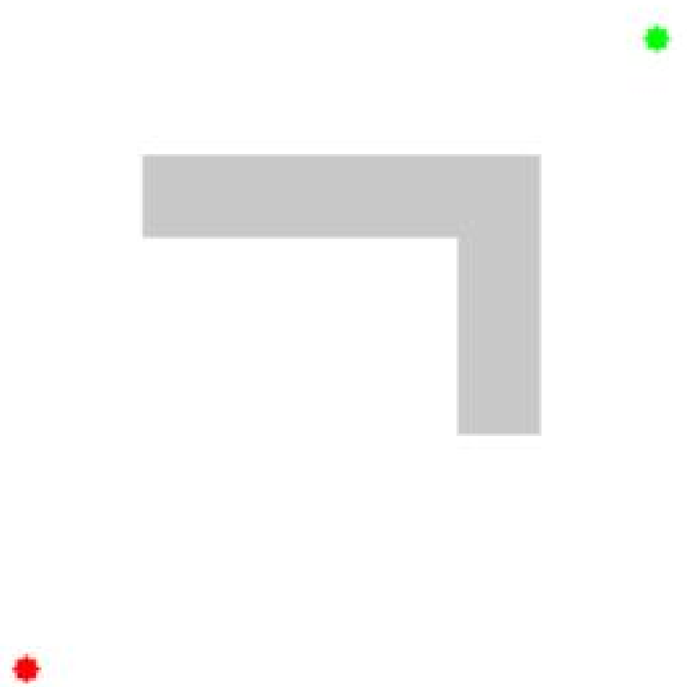
$S = \{ s, 2, 3, 4, 5, 6, 7, t \}$

$PQ = \{ \}$



Another illustration of Dijkstra's algorithm

Another illustration of Dijkstra's algorithm



Proof of Dijkstra's algorithm

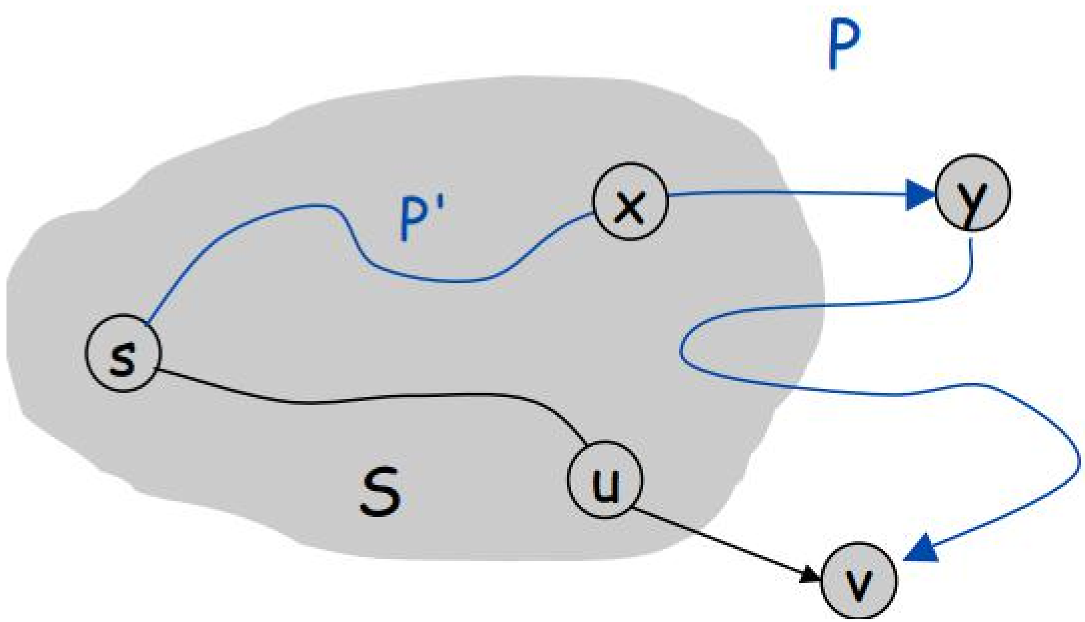
Proof of Dijkstra's algorithm

Invariant. For each node $u \in S$, $d(u)$ is the length of the shortest s-u path.

Pf. (by induction on $|S|$)

Base case: $|S| = 1$ is trivial.

Inductive hypothesis: Assume true for $|S| = k \geq 1$.



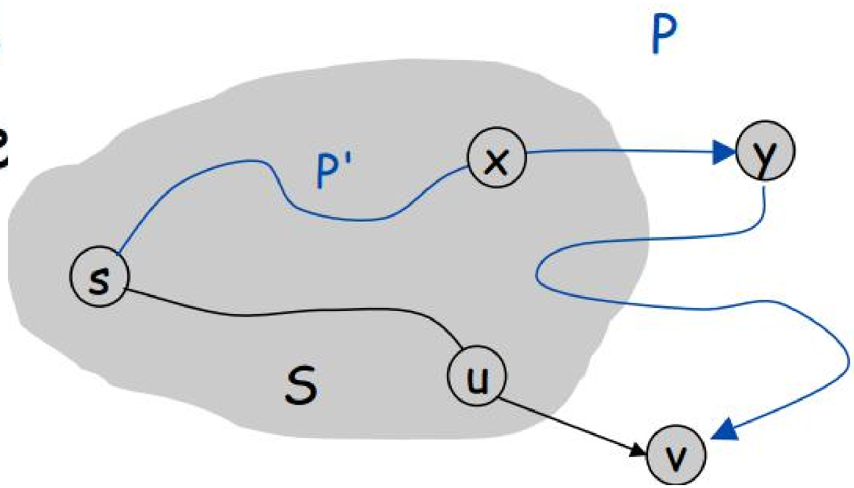
Proof of Dijkstra's algorithm

Invariant. For each node $u \in S$, $d(u)$ is the length of the shortest s-u path.

Pf. (by induction on $|S|$)

Base case: $|S| = 1$ is trivial.

Inductive hypothesis: Assume true



$$\ell(P) \geq \ell(P') + \ell(x,y) \geq d(x) + \ell(x,y) \geq \pi(y) \geq \pi(v)$$

nonnegative weights

inductive hypothesis

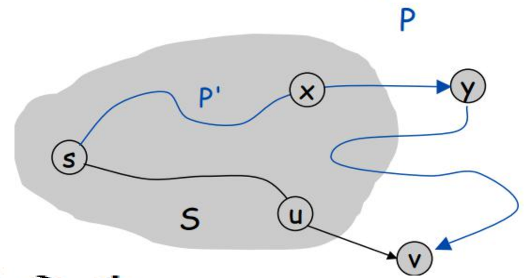
defn of $\pi(y)$

Dijkstra chose v instead of y

Proof of Dijkstra's algorithm

Invariant. For each node $u \in S$, $d(u)$ is the length of the shortest s-u path.

Pf. (by induction on $|S|$)



Base case: $|S| = 1$ is trivial.

Inductive hypothesis: Assume true for $|S| = k \geq 1$.

- Let v be next node added to S, and let u-v be the chosen edge.
- The shortest s-u path plus (u, v) is an s-v path of length $\pi(v)$.
- Consider any s-v path P. We'll see that it's no shorter than $\pi(v)$.
- Let x-y be the first edge in P that leaves S, and let P' be the subpath to x.
- P is already too long as soon as it leaves S.

$$\ell(P) \underbrace{\geq}_{\text{nonnegative weights}} \ell(P') + \ell(x,y) \underbrace{\geq}_{\text{inductive hypothesis}} d(x) + \ell(x, y) \underbrace{\geq}_{\text{defn of } \pi(y)} \pi(y) \underbrace{\geq}_{\text{Dijkstra chose v instead of y}} \pi(v)$$

讨论2：连通图有关性质

1 设G是一(n，m)图，G有ω个分图，则

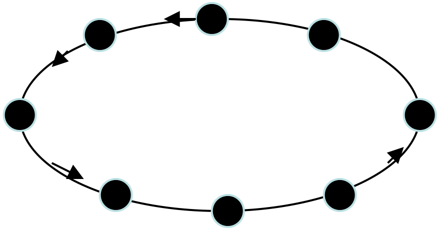
$$n-\omega \leq m \leq (n-\omega)(n-\omega+1)/2$$

2 二部图G中无长度为奇数的环。

3 有向图D强连通 $\Leftrightarrow$ D中有闭通路过每个结点至少一次.

4 有向图D单向连通 $\Leftrightarrow$ D中有通路过每个结点至少一次.

3 证明:

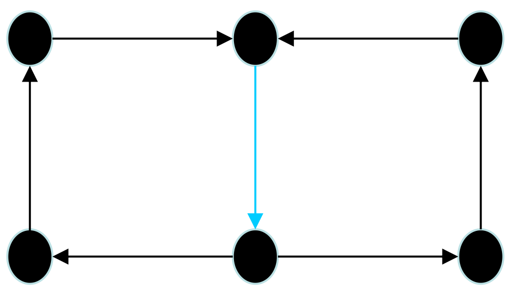


(⇐) 显然

(⇒) 设V(D)={v1,v2,…,vn}, 设$\Gamma_{i,j}$是从v_i到v_j的有向路径, 则

$\Gamma_{1,2}+\Gamma_{2,3}+\ldots+\Gamma_{n-1,n}+\Gamma_{n,1}$是过每个结点至少一次的闭通路.

思考：进一步地，一定存在满足条件的环？请举例。

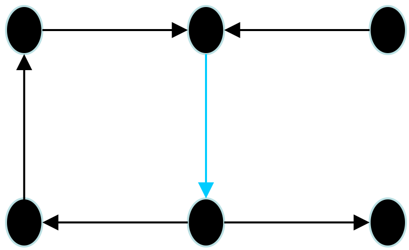


4 证明

(⇐) 显然

(⇒) 必要性证明有一定难度，……

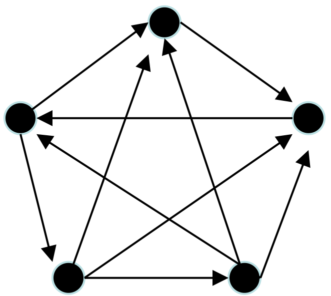
思考：一定存在满足条件的路径？请举例。



4 必要性证明

利用竞赛图相关性质 (构造法)

a 竞赛图?



引理： **竞赛图一定存在路径过每个结点恰好一次。**

如何证明上述引理?

b 根据上述引理，如何按“构造”法证明原题?

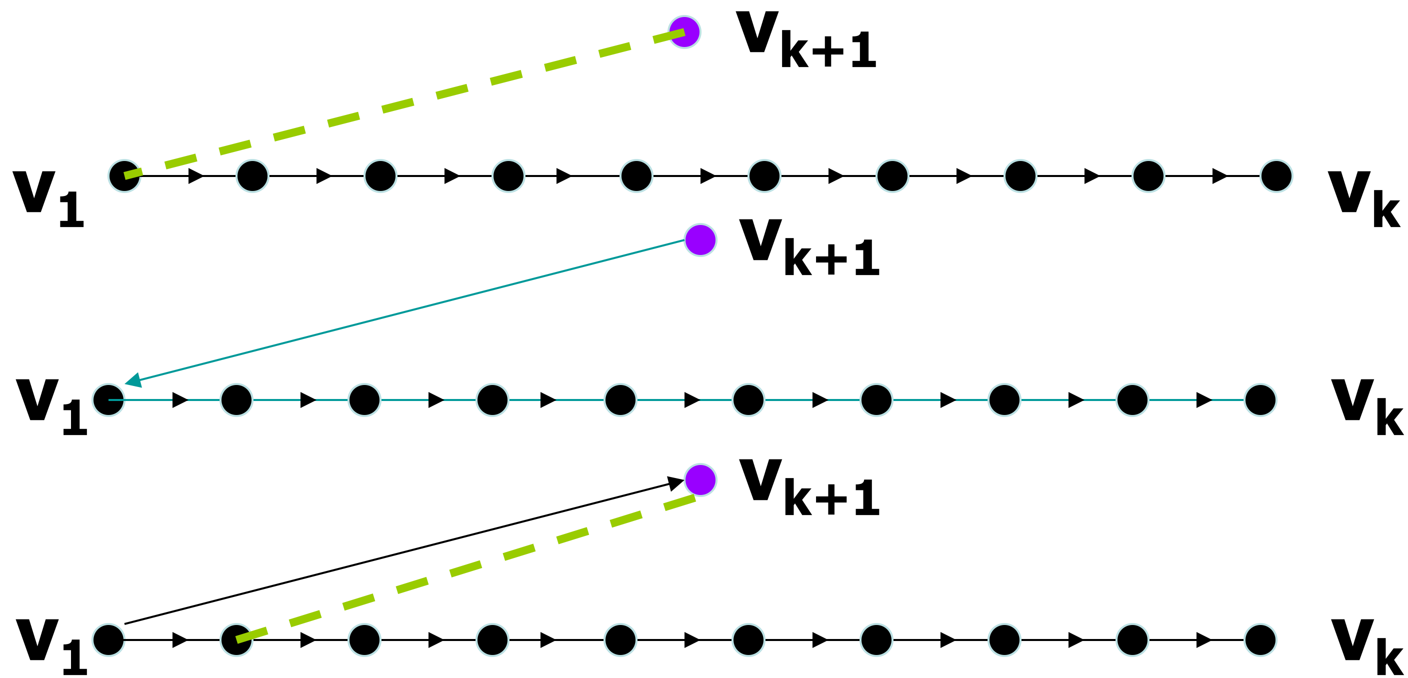
b.0 设G结点集$V=\{v_1,v_2,\ldots,v_n\}$,则在任意一对结点v_i与v_j之间至少有一条单向通路;

b.1 定义V上的竞赛图G′,使得结点v_i与v_j之间G′边的方向与G中单向通路方向一致;

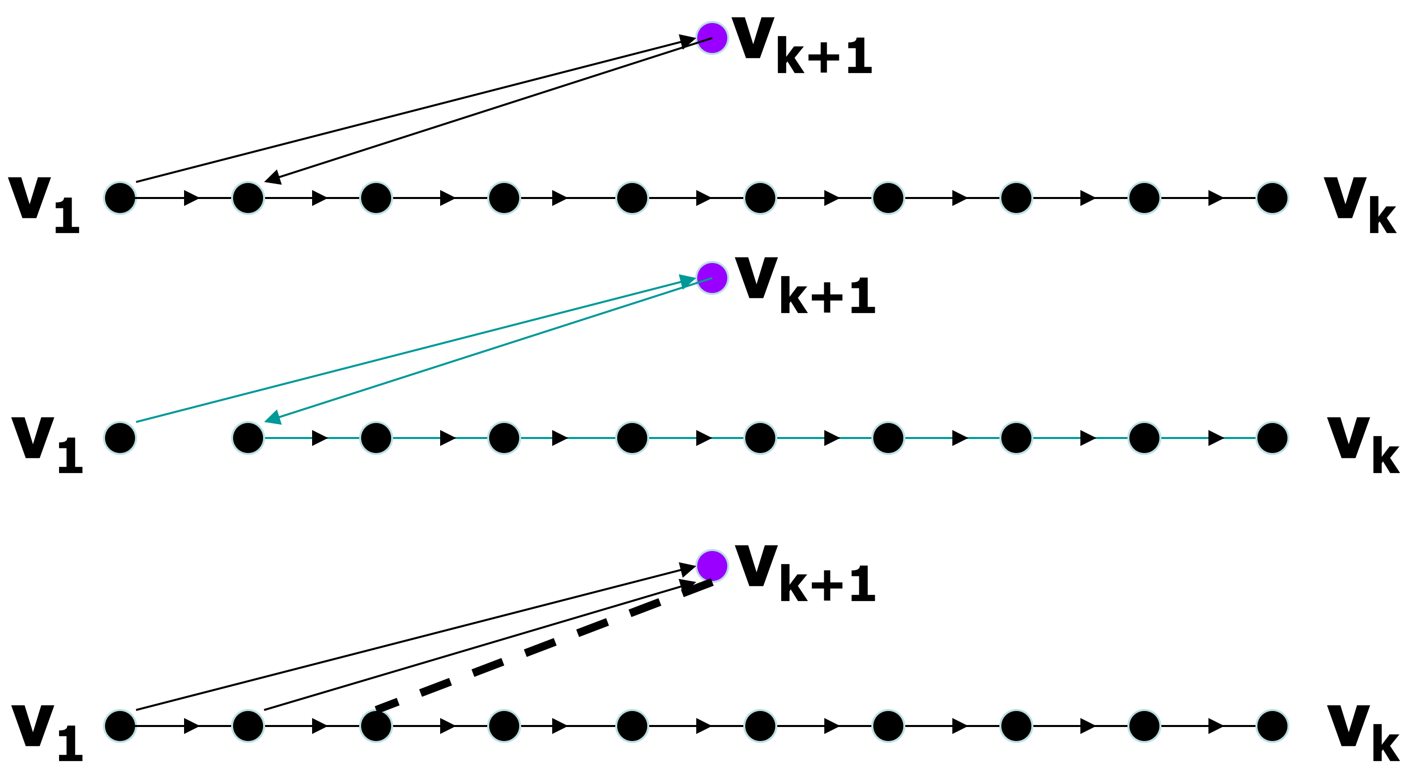
b.2 根据引理,得到G′中经过每个结点的路径;

b.3 把G中单向通路逐段“代入”上述G′中路径, 即得到所求通路.

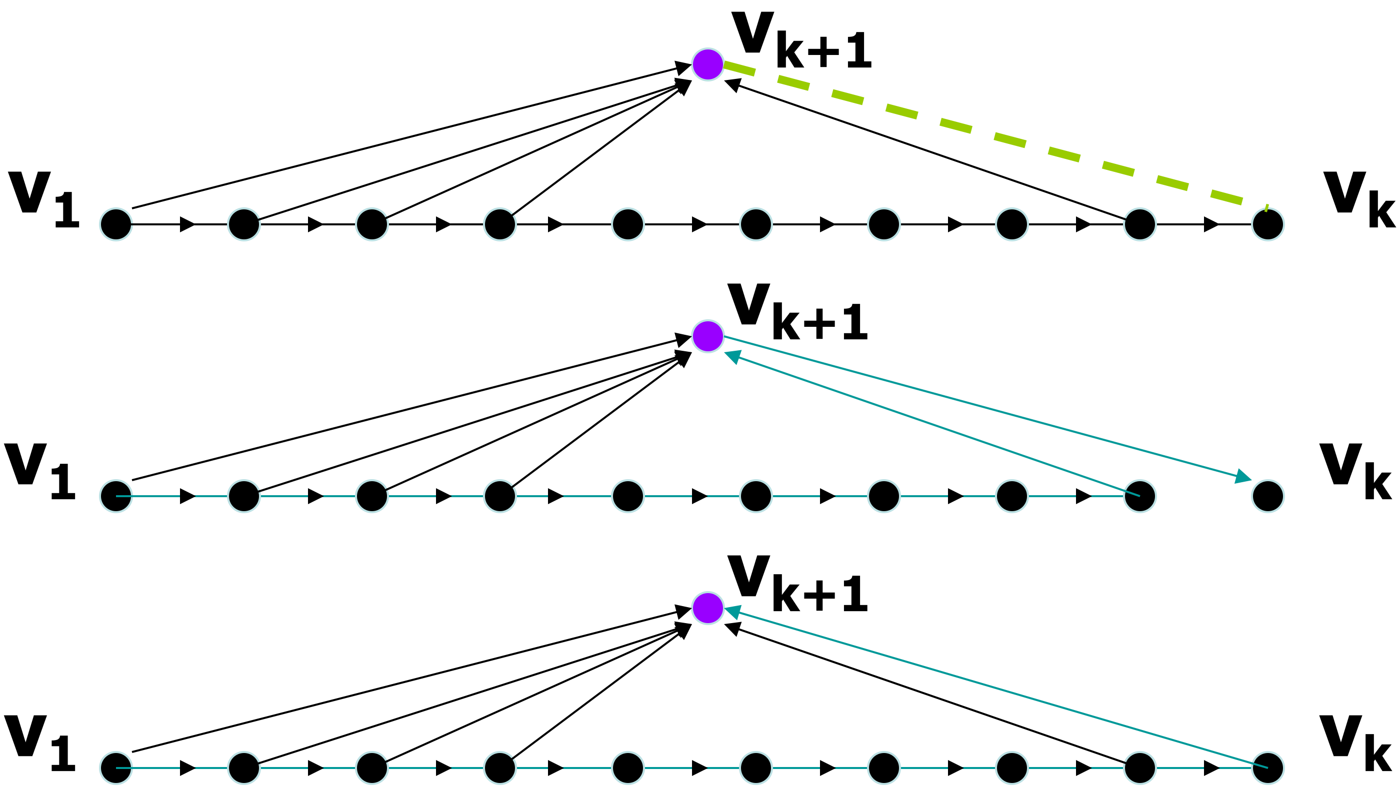
引理的证明（图示）



引理的证明（图示）



引理的证明（图示）



图的表示的三种方法

（1）集合表示

（2）邻接表（adjacency list）表示

（3）**矩阵**表示

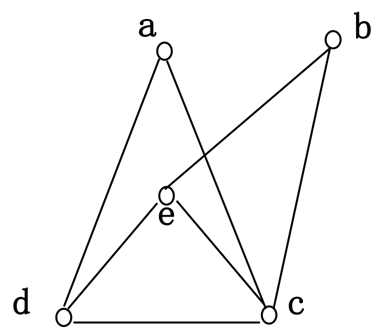
1、邻接矩阵（adjacency matrix)

2、关联矩阵（incidence matrix)

3、度数矩阵 （diagonal matrix of the degrees)

4、拉普拉斯矩阵（Laplacian matrix)

示例 邻接矩阵及其应用



$$A(G)=\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

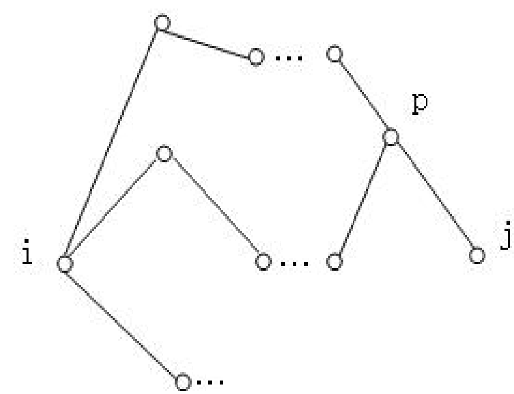
$$A(G)^2=\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ \boxed{1 \; 0 \; 1 \; 0 \; 1} \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}\begin{array}{c} \begin{array}{ccccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}=\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 1 & 1 & 1 & 2 \\ 1 & 2 & 1 & 2 & 1 \\ 1 & 1 & 4 & 2 & 2 \\ 1 & 2 & \boxed{2} & 3 & 1 \\ 2 & 1 & 2 & 1 & 2 \end{pmatrix} \end{array}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} =1\cdot1+0\cdot1+1\cdot0+0\cdot1+1\cdot1=2$$

性质 如果A是一个图G的邻接矩阵，那么A^n（n=1，2，3，…）中元素(ij)等于从结点i到结点j的长度为n的通路的数目。

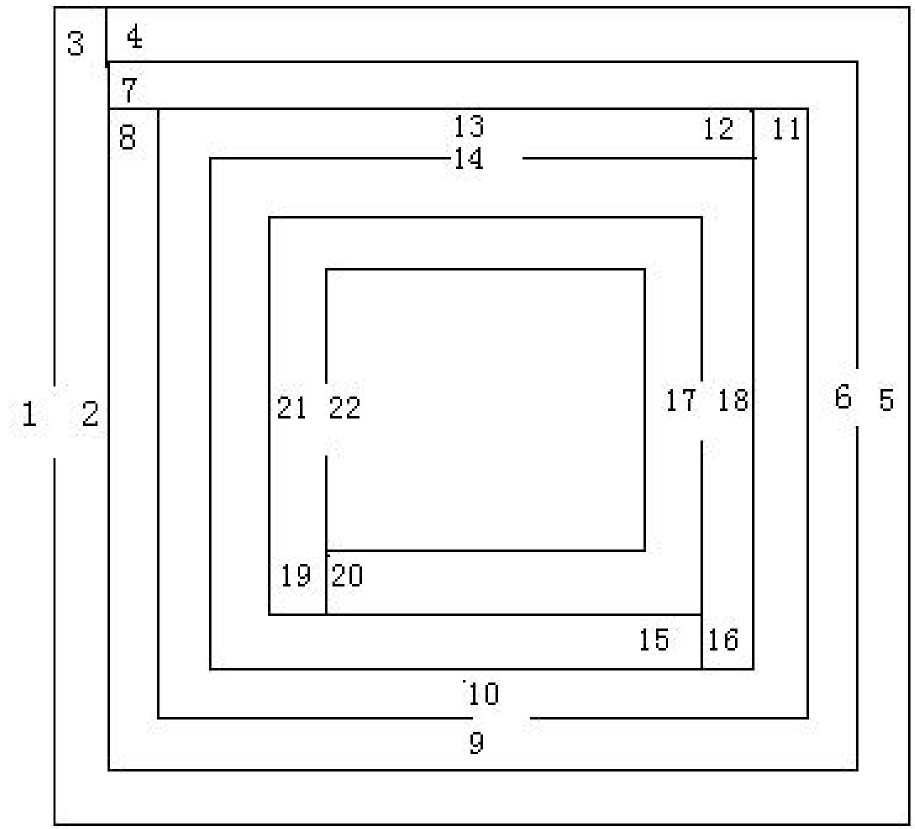
$$(ij)=A^k\text{中的第i行}\begin{pmatrix} s_1 & s_2 & \cdots & s_p & \cdots & s_m \end{pmatrix}\overset{A\text{中的第j列}}{\begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_p \\ \vdots \\ t_m \end{pmatrix}}$$

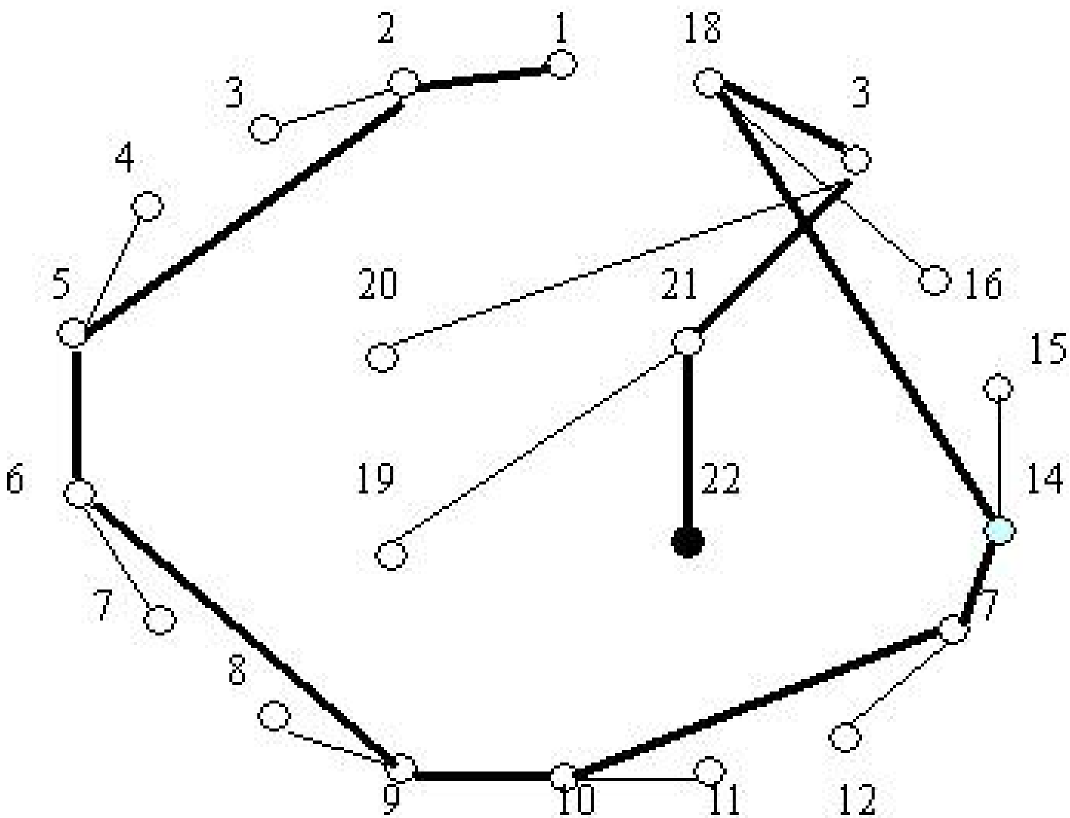
$$=s_1t_1+s_2t_2+\cdots+s_pt_p+\cdots+s_mt_m$$

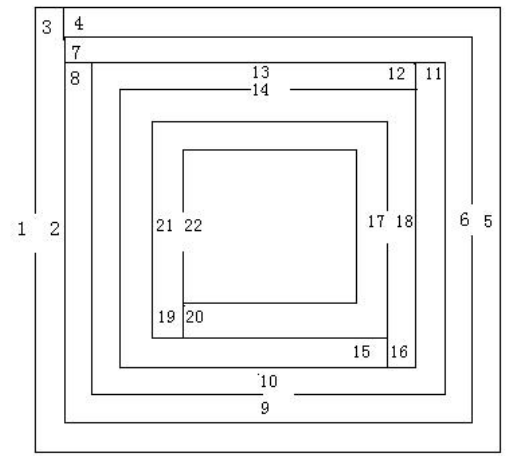


1 右图是一个迷宫,其中数字表示通道、和死胡同(包括目标)。请用一个图来表示这个迷宫(用结点表示通道、和死胡同(包括目标)),用边表示它们

之间的可直接到达关系.




2
1
18
3
3
4
5
20
21
16
15
6
19
22
14
7
7
8
9
10
11
12


3
4
7
8
13
14
12
11
1
2
21
22
17
18
6
5
19
20
15
16
10
9

2 在晚会上有n个人，他们各自与自己相识的人握一次手。已知每人与别人握手的次数都是奇数，问n是奇数还是偶数。为什么?

解：n是偶数。

用n个结点表示n个人，结点间的一条边表示一次握手，可构成一个无向图。若n是奇数，那么该图的结点度数之和为奇数个奇数的和,即为奇数，与图性质矛盾，因此，n是偶数。

3 在任何n (n≥2)个结点的简单图G中，至少有两个结点具有相同的度。

证：如果G有两个或更多孤立结点，那么它们便是具有相同的度的两个结点。如果G恰有一个孤立结点，那么我们可对有n – 1 个结点但没有孤立结点的G′(它由G删除孤立结点后得到)作讨论；如果G有分图，则也可以直接对分图进行讨论。因此，不妨设G没有孤立结点，那么G 的n个结点的度数应是：

1，2，3，…，n–1 这n–1种可能之一，

显然，必定有两个结点具有相同的度。

问题

任意一群人中至少存在两个人，他们在这群人中认识的人数恰好相等。

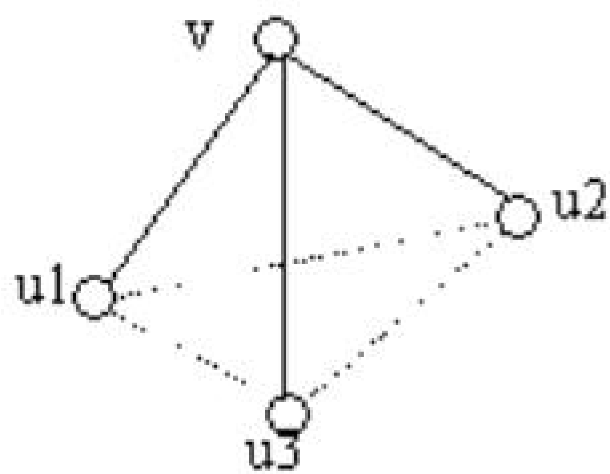
Dirichlet's drawer principle/Pigeonhole principle)：

for natural numbers k and m, if n=km+1 objects are distributed among m sets, then the pigeonhole principle asserts that at least one of the sets will contain at least k+1 objects. For arbitrary n and m this generalizes to $k+1=\lfloor (n-1)/m \rfloor + 1 = \lceil n/m \rceil$.

4 Kn表示n个结点的无向完全图。

(I) 对K_6的各边用红、蓝两色着色，每边仅着一种颜色，红、蓝任选。证明：无论怎样着色，图上总有一个红色边组成的K_3或一个蓝色边组成的K_3.

证（I）考虑K_6的结点V，与之关联的边有5条，其中至少有3条着同一颜色。不妨设均着红色，这三边的另一个端点分别是u_1,u_2,u_3 (如图所示)。再考虑关联u_1,u_2,u_3的三条边。如果它们中有一条着红色的边，那么我们就已经得到一个红色边组成的K_3，如果它们中没有着红色的边，那么我们就能够得到一个蓝色边组成的K_3



（2）用（l）证明下列事实：任意6个人之间或者有三个人相互认识，或者有3个人相互都不认识.

证明

用六个结点表示6个人，结点间红色边表示人员间相互认识，结点间蓝色边表示人员间相互不认识，便产生一个边着红、蓝两色的完全图K6。利用（1）的结论，可以断定6个人之间或者有三个人相互认识，或者有3个人相互都不认识。

Ramsey theory

The metastatement of **Ramsey theory** is that "complete disorder is impossible".

In other words, in a large system, however complicated, there is always a smaller subsystem which exhibits some sort of special structure.

Ramsey's Theorem is a species of theorem which asserts that every system of a certain type possesses a large subsystem which a higher degree of organisation than the original system.

The infinite version of Ramsey's Theory Implies that every infinite sequence contains an infinite monotone subsequence.

The metastatement of **Ramsey theory** is that "complete disorder is impossible".

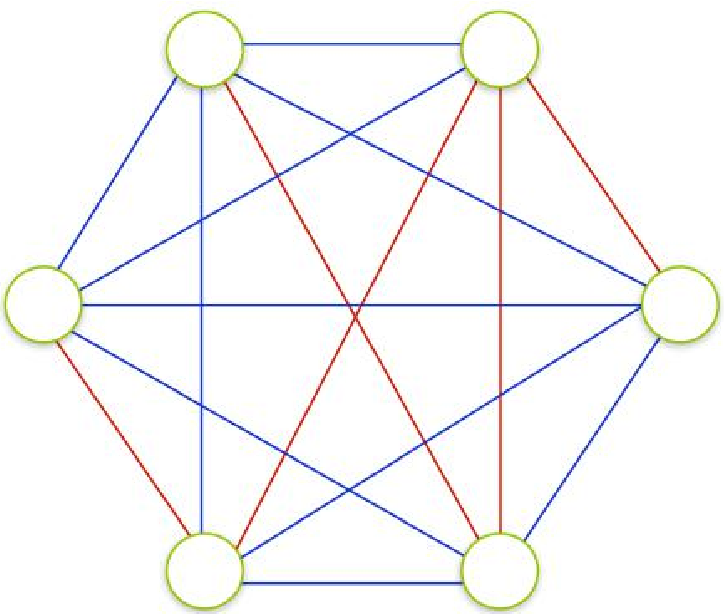
In other words, in a large system, however complicated, there is always a smaller subsystem which exhibits some sort of special structure.

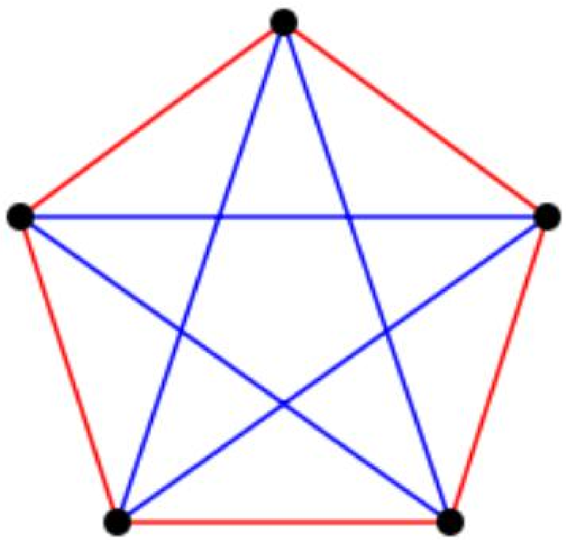
A clique of size t is a set of t vertices such that all pairs among them are edges.
An independent set of size s is a set of s vertices such that there is no edge between them.

The Ramsey number R(s, t) is the minimum number n such that any graph on n vertices contains either an independent set of size s or a clique of size t.

Ramsey' s theorem

For any s, t ≥ 1, there is $R(s, t) < \infty$ such that any graph on $R(s, t)$ vertices contains either an independent set of size s or a clique of size t.





Currently, the exact value of R(5,5) is unknown, however the best known lower and upper bounds are

43 <= R(5,5) <= 49.

But, 102<=R(6,6)<=165.

Schur's Theorem(for Fermat's Last Theorem)，Hales-Jewett Theorem, Decision Tree Complexity.
Graph theory, logic and complexity theory, combinatorics.

The old joke that is often associated with Ramsey Numbers is that if an alien spaceship were to come to Earth and demand that we tell them the answer to R(5,5) or they will kill us all, it would take all the computing power in the world to find the answer for the aliens. If he asks for R(6,6), we should try to kill them. This shows how difficult this problem is. Many people have been working on R(5,5) for decades and have made some remarkable headway.

6 问题

5 试用有向图描述下列问题的解.

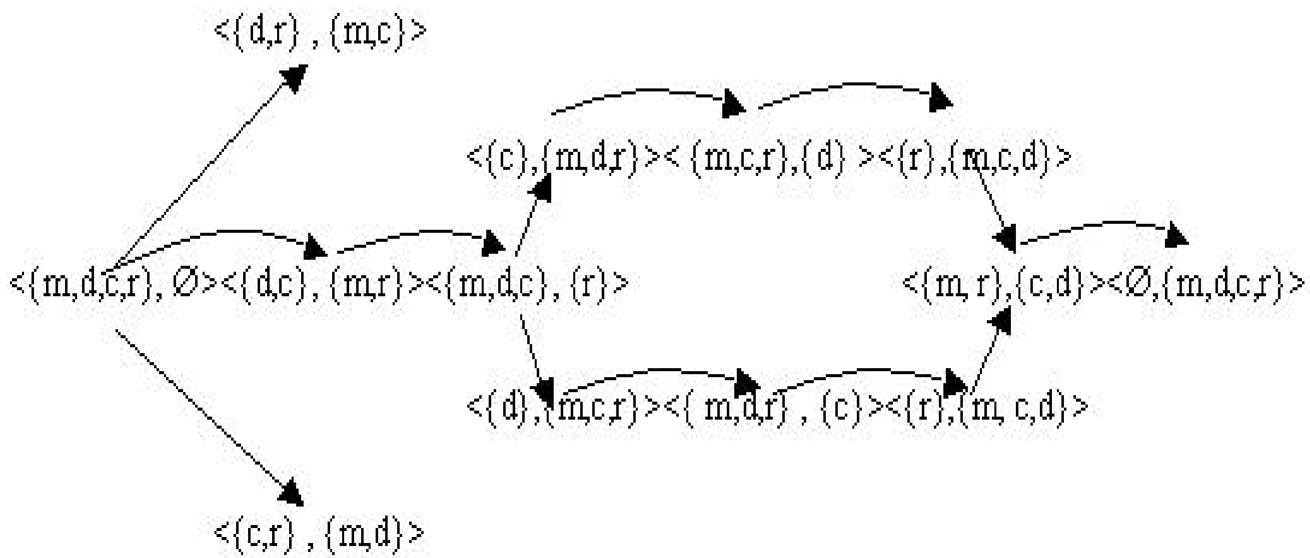
某人m带一条狗d，一只猫c和一只兔子r过河。m每次游过河时只能带一只动物，而没人管理时，狗与兔子不能共处，猫和兔子也不能共处. 问m怎样把三个动物带过河去?

提示：用结点代表状态，状态用序偶<S1，S2>来表示，这里S1，S2

分别是左岸和右岸的人及动物集合，例如初始状态为< {m,d,c,r} , ∅ >.

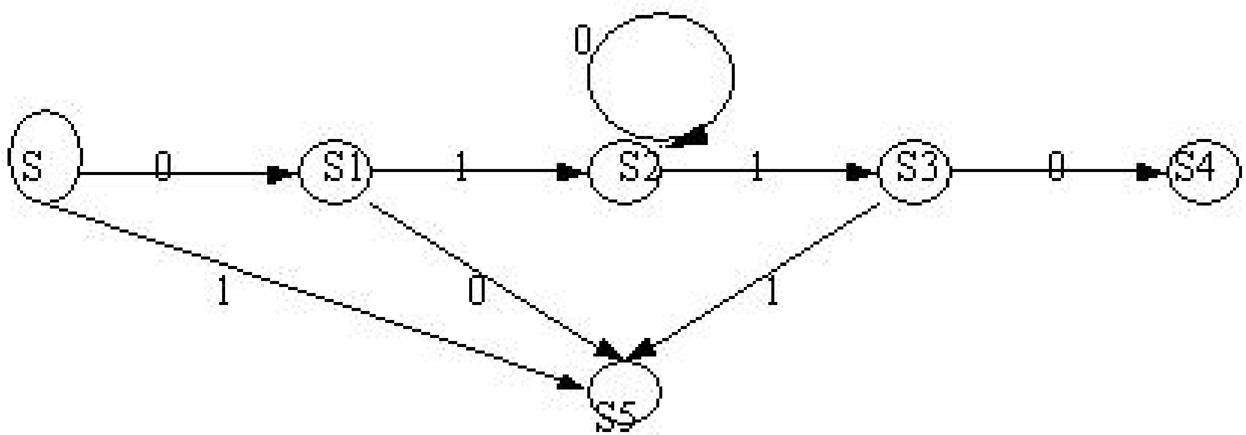
6 问题

用有向图求解

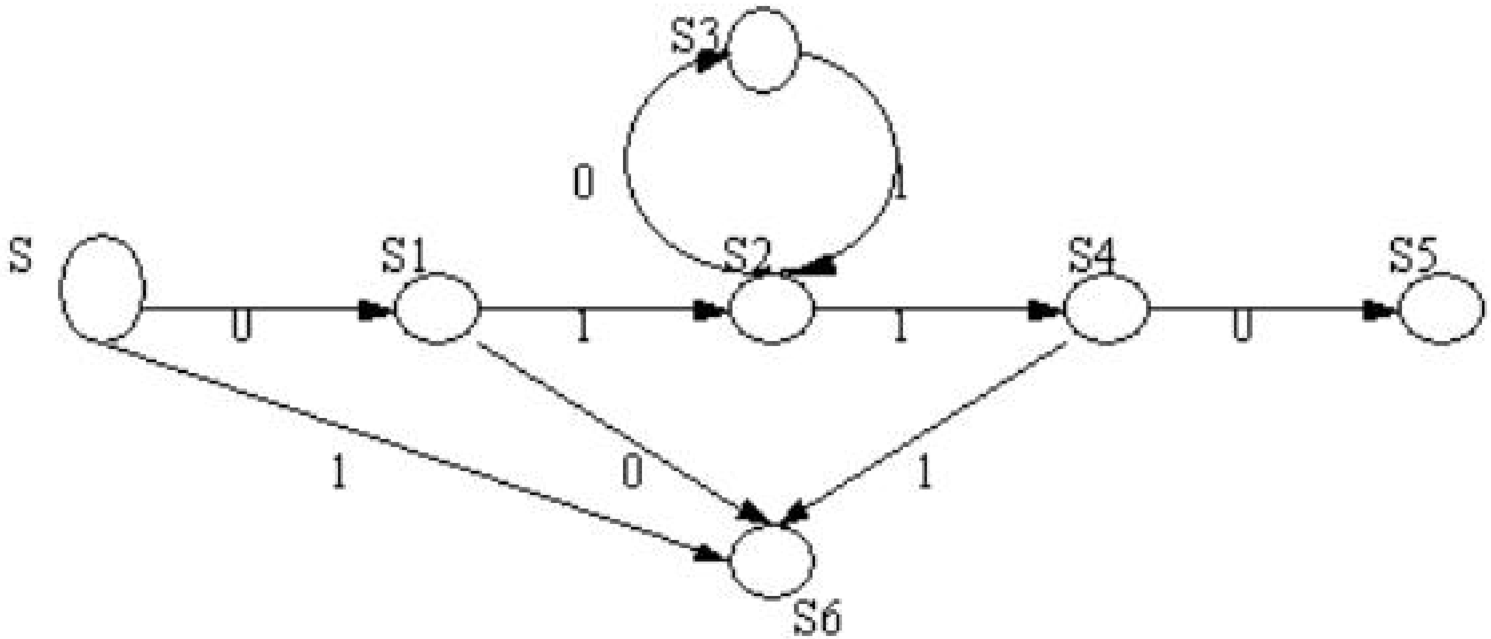


<{d,r} , {m,c}>
<{c},{m,d,r}>< {m,c,r},{d} ><{r},{m,c,d}>
<{m,d,c,r}, Ø><{d,c}, {m,r}><{m,d,c}, {r}>
<{m, r},{c,d}><Ø,{m,d,c,r}>
<{d},{m,c,r}><{ m,d,r} , {c}><{r},{m, c,d}>
<{c,r} , {m,d}>

6 有向图可以刻划一个系统的状态转换，例如用下图中的有向图可以描述识别010*10序列的状态转换系统。其中S为初始状态，在此读入序列，然后依序列中符号转入后续状态（读到0进入S1，读到1进入S2，如此等等）。S4表示读完序列010*10应进入的最后状态，S5表示读完一个非010*10序列应进入的最后状态。（上文中w*表示空字或重复任意多次w所得的字。） 试自行构作识别序列01(10)*10的有向图刻划的状态转换系统。



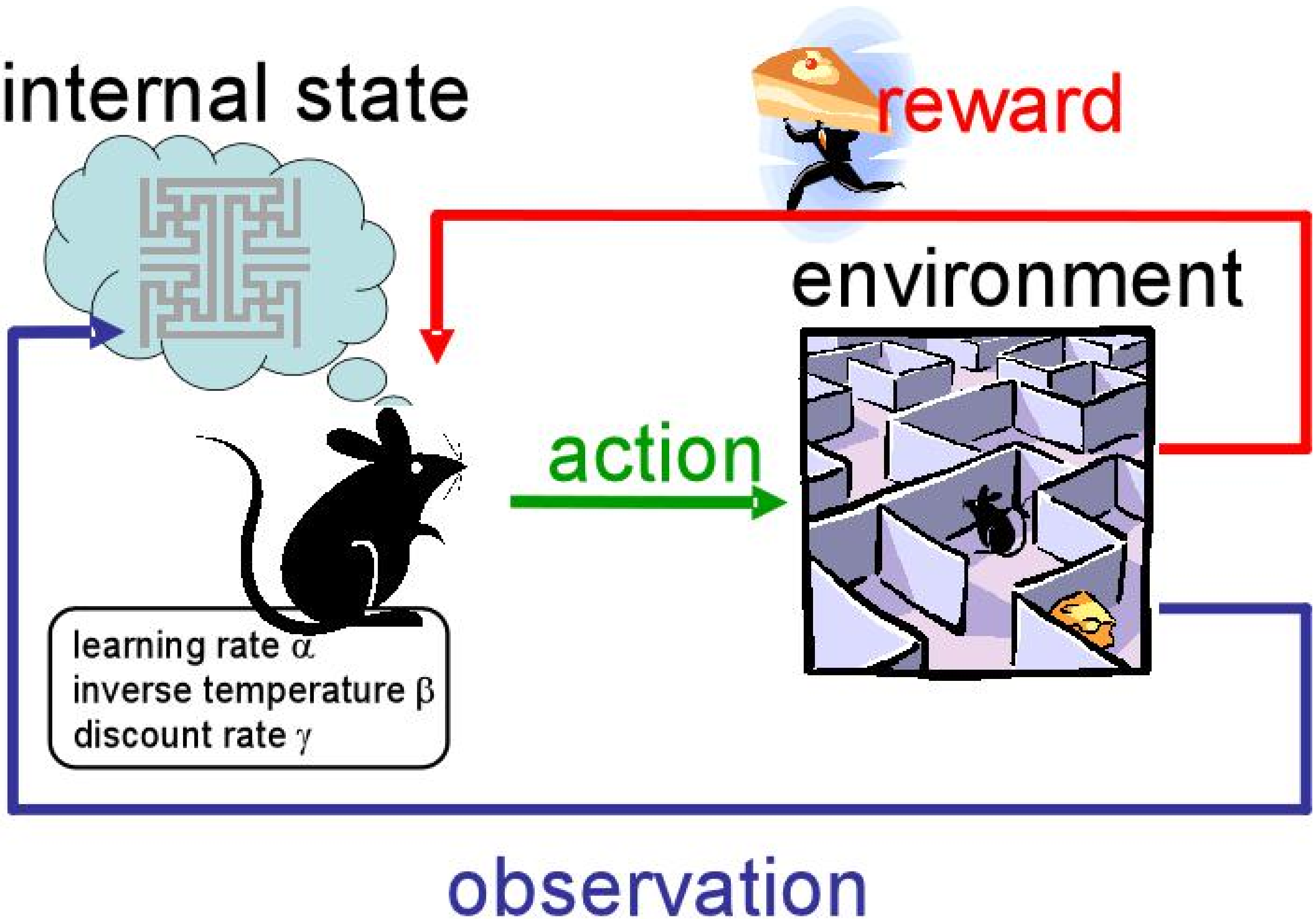
识别序列01(01)*10的有向图刻划的状态转换系统如下：



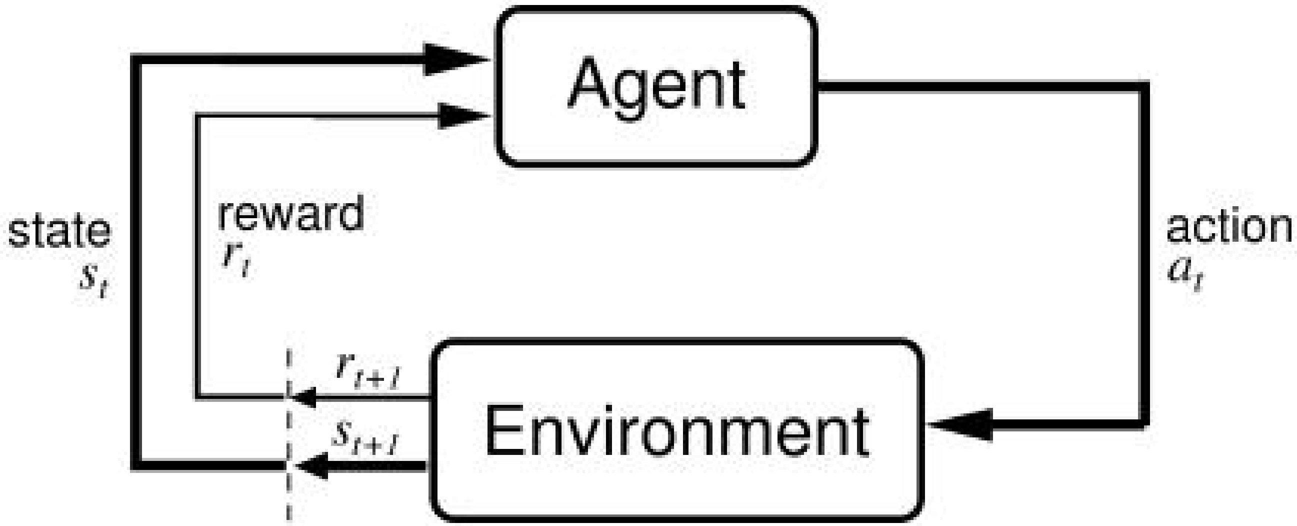
7 一个时间安排问题

学校要为一年级的研究生开设六门基础数学课：统计(S),数值分析(N)，图论(G),矩阵论（M）,随机过程(R)和数理方程(P)。按培养计划，注册的学生必须选修其中的一门以上。你作为教务管理人员，要设法安排一个课表，使每个学生所选的课程，在时间上不会发生冲突．由于他们都要必修的外语和自然辨证法课程占用了许多时间，可供排课的时段不多，可供使用的教室足够多，因此，有些课可能需要安排在同样的时段。上述六门数学课中任何两门，只要不被同一个学生选择，可以安排在同样的时间段。

8 DQN and Q-Learning

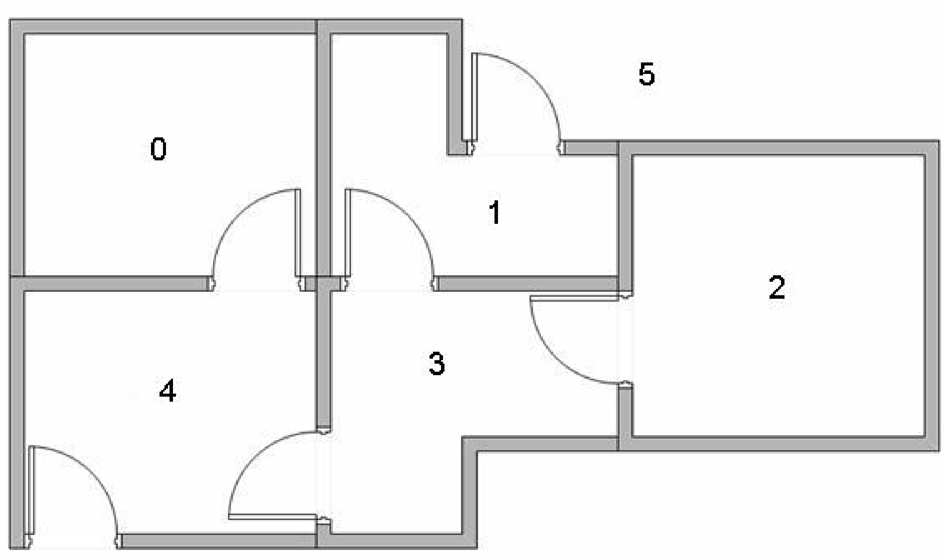


8 DQN and Q-Learning



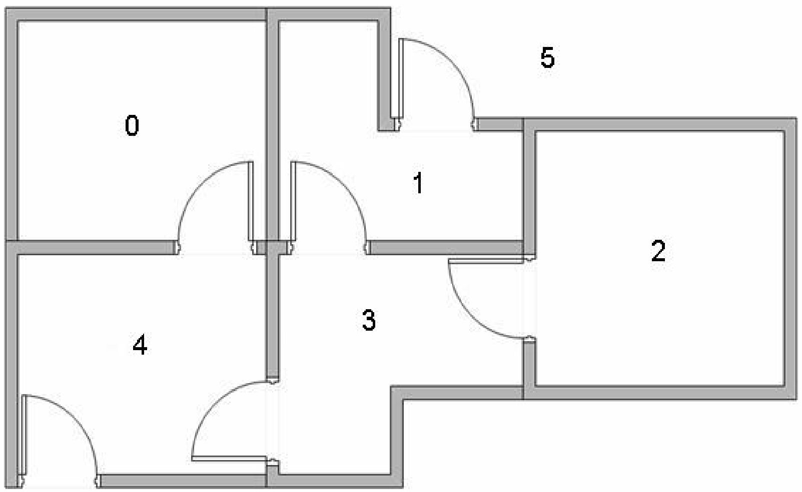
8 DQN and Q-Learning

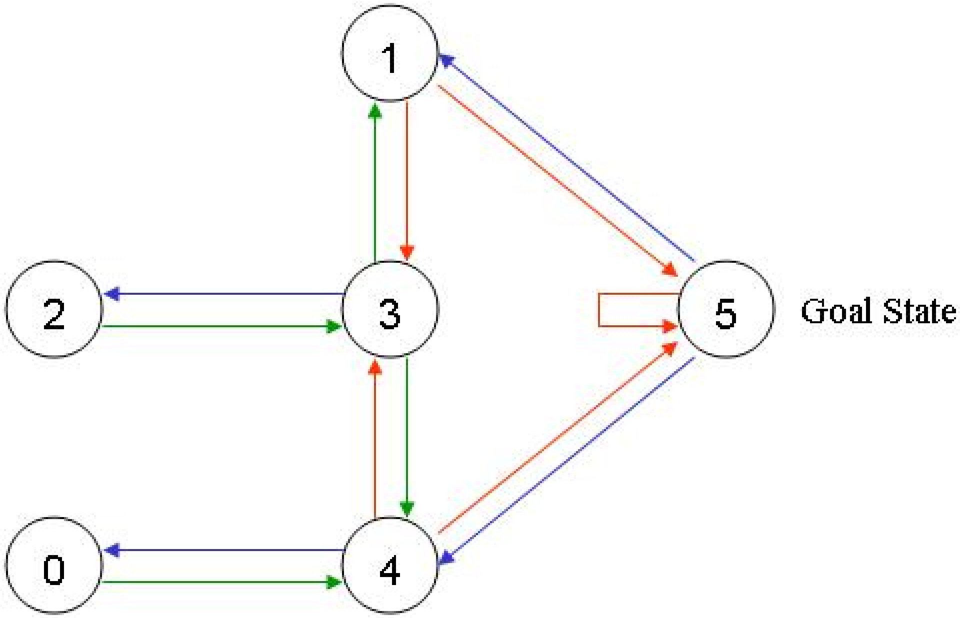
Suppose we have 5 rooms in a building connected by doors as shown in the figure below. We'll number each room 0 through 4. The outside of the building can be thought of as one big room (5). Notice that doors 1 and 4 lead into the building from room 5 (outside).We'd like to put an agent in any room, and from that room, go outside the building (this will be our target room).



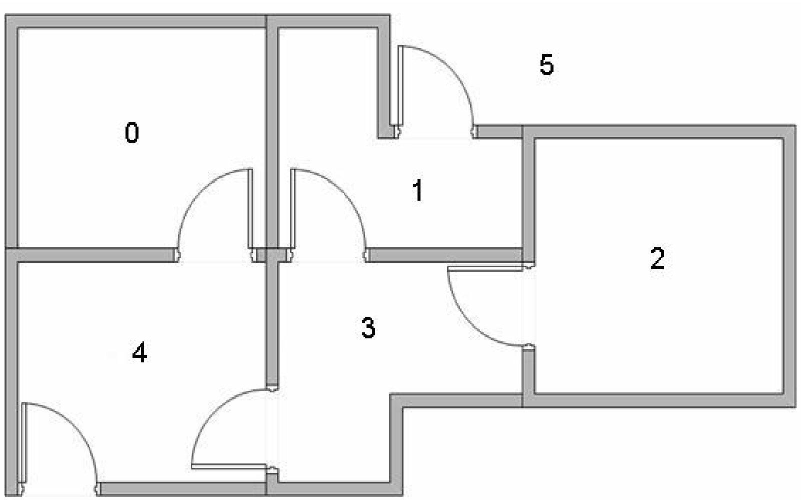
from http://mnemstudio.org/path-finding-q-learning.htm

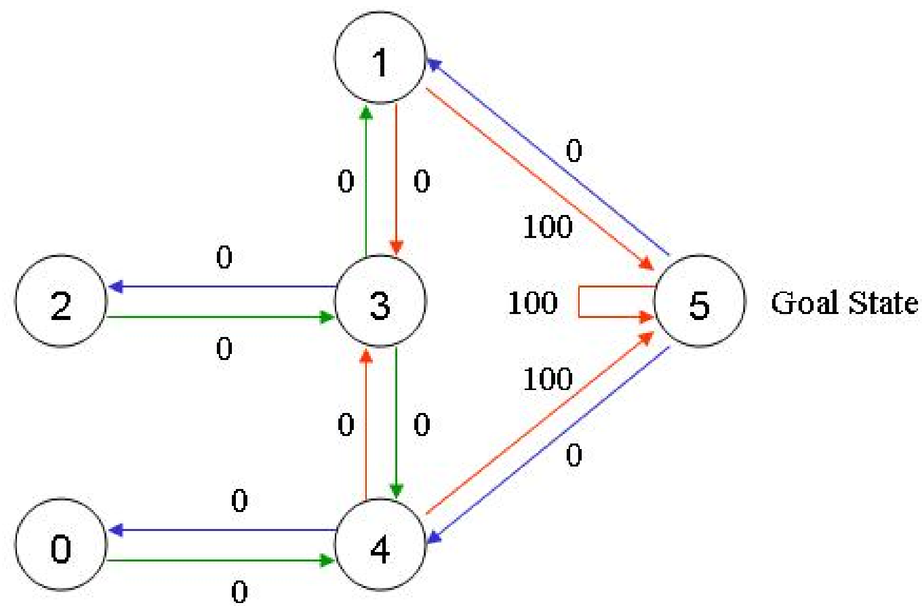
Imagine our agent as a dumb virtual robot that can learn through experience. The agent can pass from one room to another but has no knowledge of the environment, and doesn't know which sequence of doors lead to the outside. Suppose we want to model some kind of simple evacuation of an agent from any room in the building.



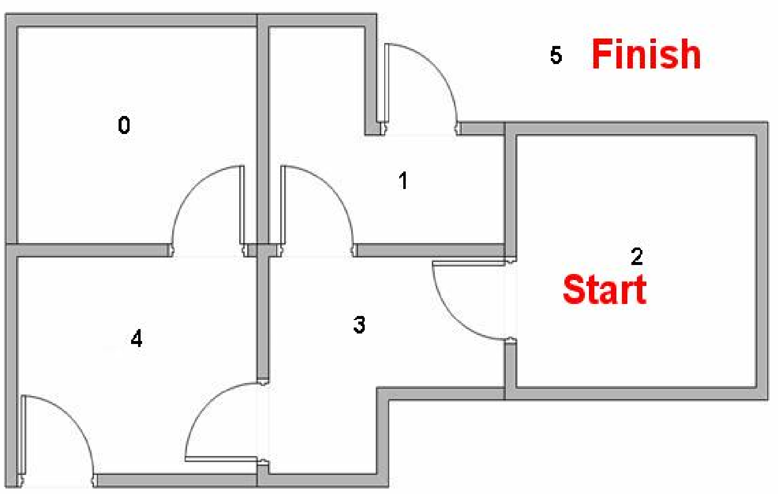


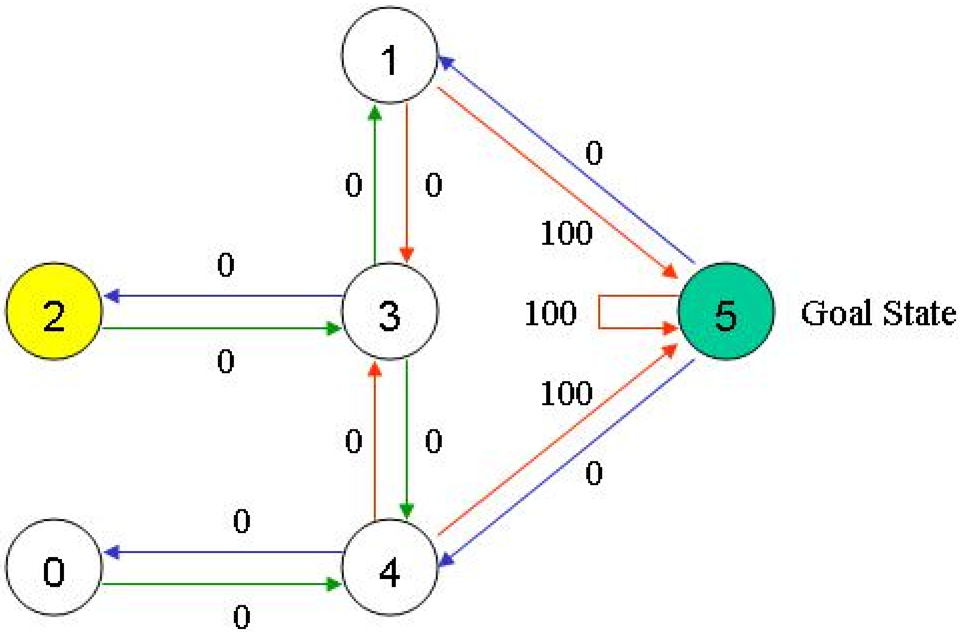
Imagine our agent as a dumb virtual robot that can learn through experience. The agent can pass from one room to another but has no knowledge of the environment, and doesn't know which sequence of doors lead to the outside. Suppose we want to model some kind of simple evacuation of an agent from any room in the building.

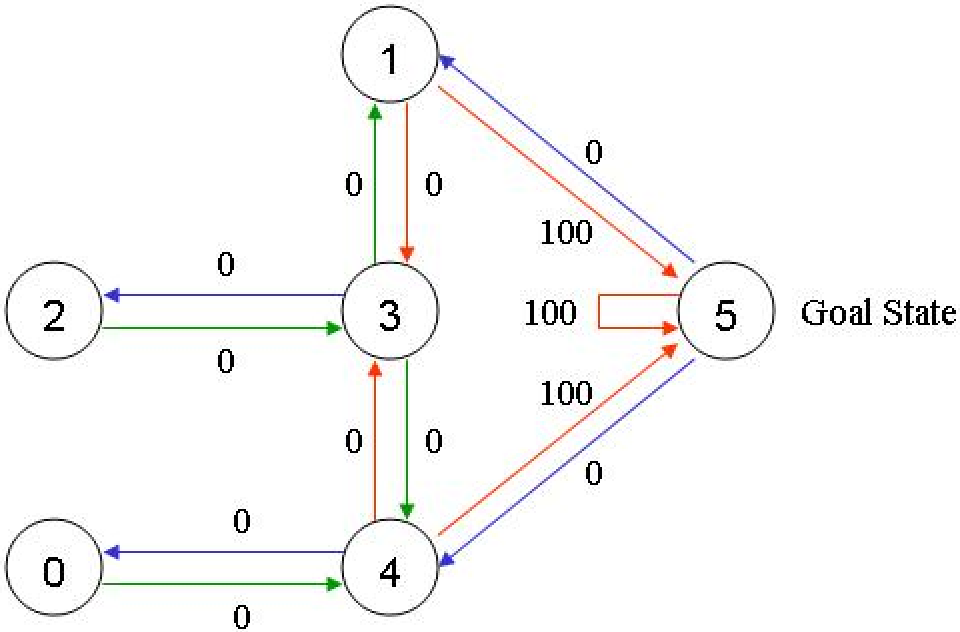




Imagine our agent as a dumb virtual robot that can learn through experience. The agent can pass from one room to another but has no knowledge of the environment, and doesn't know which sequence of doors lead to the outside. Suppose we want to model some kind of simple evacuation of an agent from any room in the building.







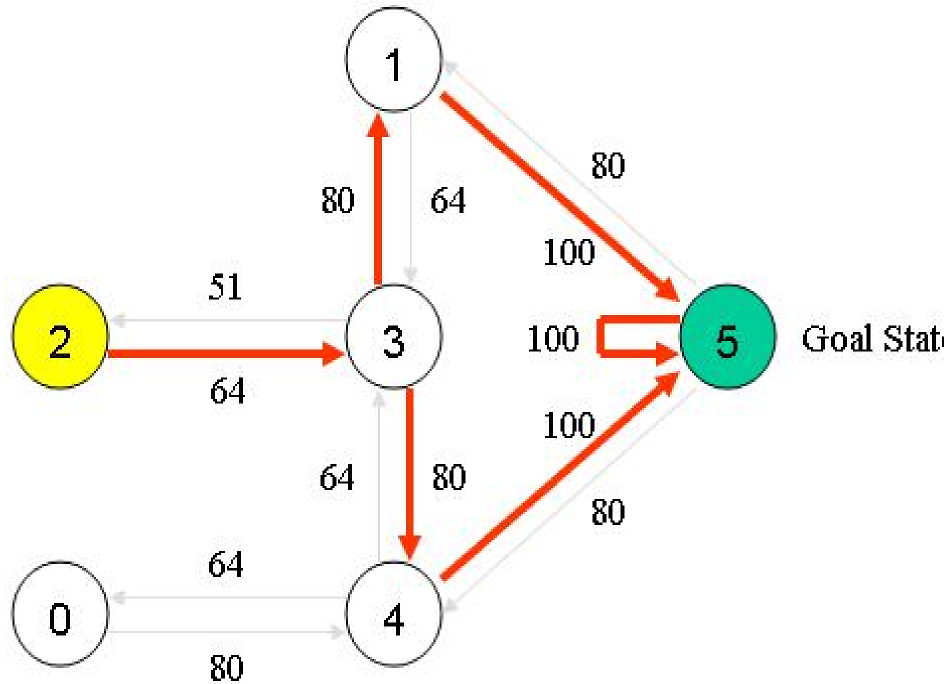
Reward Table

$$
R=\begin{array}{c} \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
-1 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & -1 & 0 & -1 & 100 \\
-1 & -1 & -1 & 0 & -1 & -1 \\
-1 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & -1 & 0 & -1 & 100 \\
-1 & 0 & -1 & -1 & 0 & 100
\end{bmatrix}
\end{array}
$$

Q Table

$$
Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

$$
Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 80 & 0 \\
0 & 0 & 0 & 64 & 0 & 100 \\
0 & 0 & 0 & 64 & 0 & 0 \\
0 & 80 & 51 & 0 & 80 & 0 \\
64 & 0 & 0 & 64 & 0 & 100 \\
0 & 80 & 0 & 0 & 80 & 100
\end{bmatrix}
\end{array}
$$



Shortest routes from initial states: 2 - 3 - 1 - 5.