- 作业讲解
  - TC第25.1节练习4、5、6、9、10
  - TC第25.2节练习2、4、6、8
  - TC第25.3节练习2、3
  - TC第25章问题2

TC第25.1节练习4

25.1-4
Show that matrix multiplication defined by EXTEND-SHORTEST-PATHS is associative.

EXTEND-SHORTEST-PATHS(L, W)

```
n = L.rows
let L' = (l'_ij) be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        l'_ij = ∞
        for k = 1 to n
            l'_ij = min(l'_ij, l_ik + w_kj)
return L'
```

$$
\begin{array}{rcl}
l^{(m-1)} & \rightarrow & a\,, \\
w & \rightarrow & b\,, \\
l^{(m)} & \rightarrow & c\,, \\
\min & \rightarrow & +\,, \\
+ & \rightarrow & \cdot
\end{array}
$$

TC第25.1节练习5

25.1-5
Show how to express the single-source shortest-paths problem as a product of matrices and a vector. Describe how evaluating this product corresponds to a Bellman-Ford-like algorithm (see Section 24.1).

```
BELLMAN-FORD(G, w, s)
  INITIALIZE-SINGLE-SOURCE(G, s)
  for i = 1 to |G.V| - 1
      for each edge (u, v) ∈ G.E
          RELAX(u, v, w)
```

```
RELAX(u, v, w)
  if v.d > u.d + w(u, v)
      v.d = u.d + w(u, v)
      v.π = u
```

```
EXTEND-SHORTEST-PATHS(L, W)
  n = L.rows
  let L' = (l'_ij) be a new n × n matrix
  for i = 1 to n
      for j = 1 to n
          l'_ij = ∞
          for k = 1 to n
              l'_ij = min(l'_ij, l_ik + w_kj)
  return L'
```

TC第25.1节练习6

25.1-6

Suppose we also wish to compute the vertices on shortest paths in the algorithms of this section. Show how to compute the predecessor matrix Π from the completed matrix L of shortest-path weights in $O(n^3)$ time.

for i…

  for j…

    for k…

      if (L_{ij}==L_{ik}+W_{kj})

      then Π_{ij}=k

可不可以改成

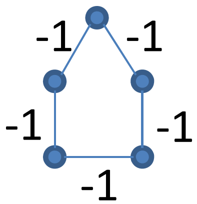
      if (L_{ij}==L_{ik}+L_{kj})

      then Π_{ij}=Π_{kj}

TC第25.1节练习9

25.1-9
Modify FASTER-ALL-PAIRS-SHORTEST-PATHS so that it can determine whether the graph contains a negative-weight cycle.

- 直接检查结果对角线上是否有负值，行不行？
  - n=5, m=4时，运行结束



FASTER-ALL-PAIRS-SHORTEST-PATHS(W)

```
n = W.rows
L^(1) = W
m = 1
while m < n - 1
    let L^(2m) be a new n × n matrix
    L^(2m) = EXTEND-SHORTEST-PATHS(L^(m), L^(m))
    m = 2m
return L^(m)
```

TC第25.2节练习4

25.2-4

As it appears above, the Floyd-Warshall algorithm requires $\Theta(n^3)$ space, since we compute $d_{ij}^{(k)}$ for $i, j, k = 1, 2, \ldots, n$. Show that the following procedure, which simply drops all the superscripts, is correct, and thus only $\Theta(n^2)$ space is required.

```
FLOYD-WARSHALL'(W)
n = W.rows
D = W
for k = 1 to n
    for i = 1 to n
        for j = 1 to n
            d_ij = min(d_ij, d_ik + d_kj)
return D
```

TC第25.3节练习2

25.3-2

What is the purpose of adding the new vertex s to V, yielding V'?

- 最直接的原因
  - 每个起点未必都可达负权圈

JOHNSON(G, w)

```
compute G', where G'.V = G.V ∪ {s},
      G'.E = G.E ∪ {(s, v) : v ∈ G.V}, and
      w(s, v) = 0 for all v ∈ G.V
if BELLMAN-FORD(G', w, s) == FALSE
      print "the input graph contains a negative-weight cycle"
else for each vertex v ∈ G'.V
            set h(v) to the value of δ(s, v)
                  computed by the Bellman-Ford algorithm
      for each edge (u, v) ∈ G'.E
            ŵ(u, v) = w(u, v) + h(u) − h(v)
      let D = (d_uv) be a new n × n matrix
      for each vertex u ∈ G.V
            run DIJKSTRA(G, ŵ, u) to compute δ̂(u, v) for all v ∈ G.V
            for each vertex v ∈ G.V
                  d_uv = δ̂(u, v) + h(v) − h(u)
      return D
```

TC第25.3节练习3

25.3-3
Suppose that $w(u,v) \geq 0$ for all edges $(u,v) \in E$. What is the relationship between the weight functions w and $\hat{w}$?

1. 对任意顶点：δ=0
2. h=δ=0
3. $\hat{w}(u,v)=w(u,v)+h(u)-h(v)=w(u,v)$

TC第25章问题2

25-2 *Shortest paths in ϵ-dense graphs*

A graph $G = (V, E)$ is ***ϵ-dense*** if $|E| = \Theta(V^{1+\epsilon})$ for some constant ϵ in the range $0 < \epsilon \leq 1$. By using d-ary min-heaps (see Problem 6-2) in shortest-paths algorithms on ϵ-dense graphs, we can match the running times of Fibonacci-heap-based algorithms without using as complicated a data structure.

a. What are the asymptotic running times for INSERT, EXTRACT-MIN, and DECREASE-KEY, as a function of d and the number n of elements in a d-ary min-heap? What are these running times if we choose $d = \Theta(n^{\alpha})$ for some constant $0 < \alpha \leq 1$? Compare these running times to the amortized costs of these operations for a Fibonacci heap.

- INSERT: 代价=树高=$\log_d n$=1/α
- DECREASE-KEY: 代价=INSERT=$\log_d n$=1/α
- EXTRACT-MIN: 代价=分支因子*树高=$d\log_d n$=n^{α}/α

TC第25章问题2 (续)

b. Show how to compute shortest paths from a single source on an ϵ-dense directed graph $G = (V, E)$ with no negative-weight edges in $O(E)$ time. (*Hint:* Pick d as a function of ϵ.)

- V次EXTRACT-MIN + E次DECREASE-KEY
- 取$d=V^{\varepsilon}$： $O(V*V^{\varepsilon}/\varepsilon + E*1/\varepsilon) = O(V^{1+\varepsilon}/\varepsilon+V^{1+\varepsilon}/\varepsilon) = O(V^{1+\varepsilon}) = O(E)$

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

TC第25章问题2 (续)

c. Show how to solve the all-pairs shortest-paths problem on an ϵ-dense directed graph $G = (V, E)$ with no negative-weight edges in $O(VE)$ time.

- (b)运行V次：V*O(E) = O(VE)

TC第25章问题2 (续)

d. Show how to solve the all-pairs shortest-paths problem in $O(VE)$ time on an ϵ-dense directed graph $G = (V, E)$ that may have negative-weight edges but has no negative-weight cycles.

- Johnson算法：Bellman-Ford + V次Dijkstra
- O(VE) + O(VE) = O(VE)

- 教材讨论
  - GC第7章

问题1：有向图的基本概念

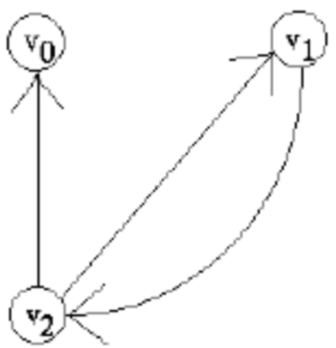
- 你能举出有向图在实际生活中的至少4个例子吗？
- 与无向图相比，这些概念在有向图中发生了怎样的变化？
  - edge、degree
  - (closed) walk、(closed) trail、path/cycle、distance
  - connectivity

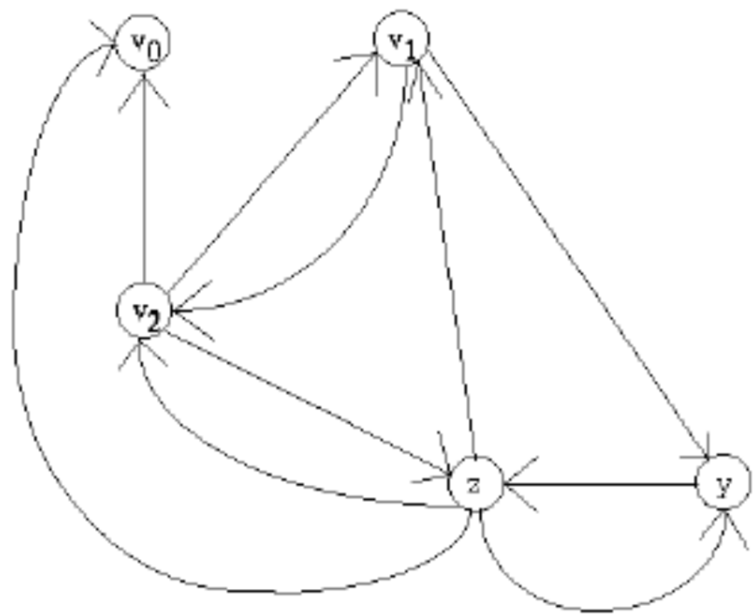
问题2：有向图的度

- 有向图中，所有顶点的入度和等于出度和吗？
- 简单有向图中，顶点的出度有可能两两互不相同吗？
  在此基础上，顶点的入度有可能却都相同吗？
  如果有可能，你能造出多少个这样的图？

问题2：有向图的度

- 有向图中，所有顶点的入度和等于出度和吗？
- 简单有向图中，顶点的出度有可能两两互不相同吗？
  在此基础上，顶点的入度有可能却两两相同吗？
  如果有可能，你能造出多少个这样的图？

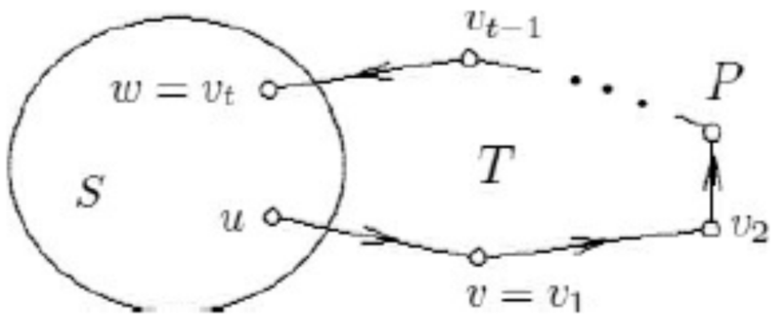




问题3：图的定向

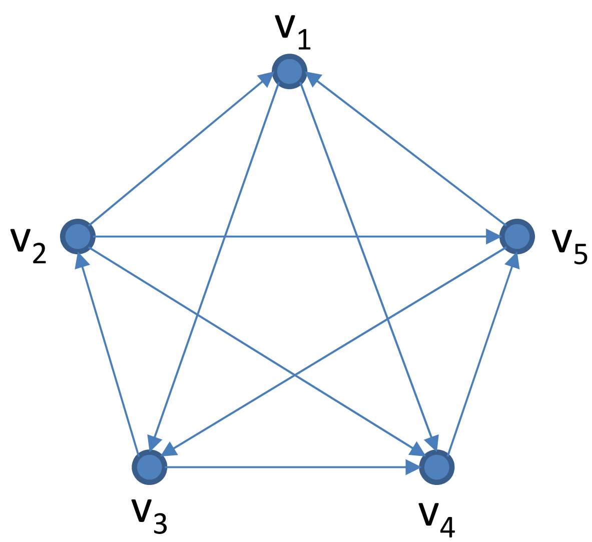
- 什么是定向(orientation)和底图(underlying graph)？
- 你能结合这个图简要证明强定向的充要条件吗？

Theorem 7.5 *A nontrivial connected graph G has a strong orientation if and only if G contains no bridges.*



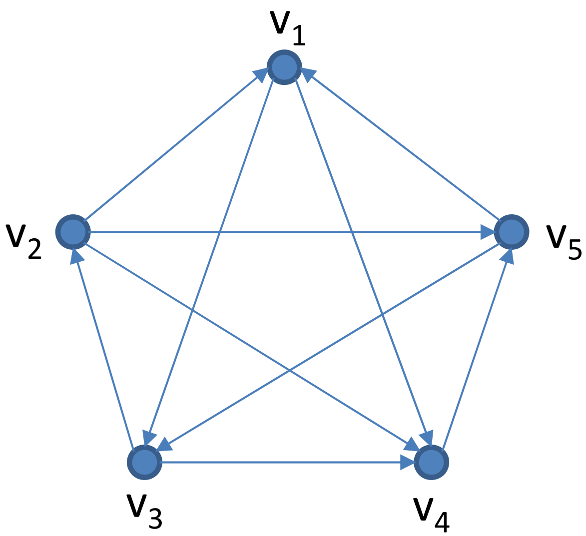
问题4：竞赛图

- 什么是竞赛图？
- 你能想到哪些方法来确定竞赛的胜者？



问题4：竞赛图 (续)

- 竞赛图中的王(king)
  - 到其它任何顶点都有长不超过2的有向路
- 王唯一吗？
- 王的充分条件（暨存在性）：出度最大
  - 你能证明吗？
  - 这同时也是必要条件吗？



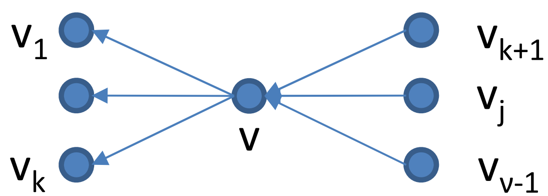
问题4：竞赛图 (续)

- 竞赛图中出度最大的顶点必为王。

证明：

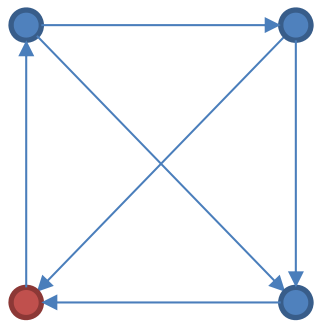
设v是出度最大的顶点。

- 如果$d^+(v)=v-1$：显然成立。
- 如果$d^+(v)<v-1$，设v的出邻点为$v_1,\ldots,v_k$，入邻点为$v_{k+1},\ldots,v_{v-1}$
  - 对于$v_{k+1},\ldots,v_{v-1}$中的每个v_j：
    $d^+(v_j)\le d^+(v) \Rightarrow v_1,\ldots,v_k$不可能都是$v_j$的出邻点（为什么？）
    $\Rightarrow$ 其中某个是v_j的入邻点 $\Rightarrow$ 从v到v_j有长为2的有向路 $\Rightarrow$ 得证



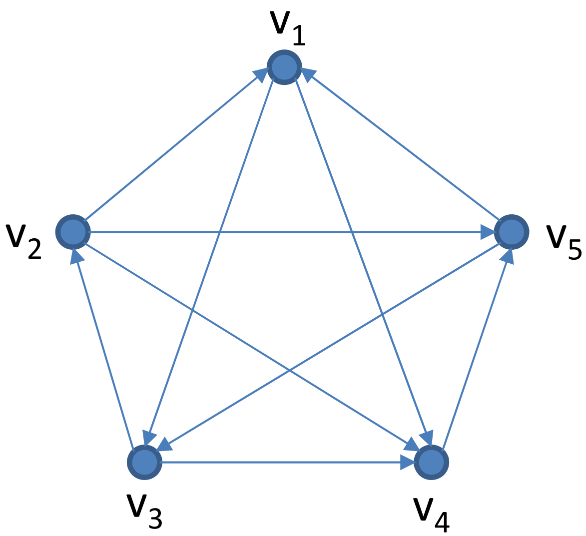
问题4：竞赛图 (续)

- 竞赛图中出度非最大的顶点也可能为王。



问题4：竞赛图 (续)

- 竞赛图中的王(king)
  - 到其它任何顶点都有长不超过2的有向路
- 王唯一吗？
- 王的充分条件（暨存在性）：出度最大
  - 你能证明吗？
  - 这同时也是必要条件吗？
- 你能为王的唯一性找一个充要条件吗？



问题4：竞赛图 (续)

- 竞赛图中一个顶点v是唯一的王当且仅当v的出度为v-1。

证明：

⇒：反证法

1. 假设唯一的王v满足$d^+(v)<v-1$ ⇒ v的所有入邻点导出的子竞赛图有自己的王u
2. u到v有弧 ⇒ u到v的出邻点有长为2的有向路

⇒ u也是原图的王 ⇒ v不是唯一的王 ⇒ 矛盾

⇐：$d^+(v)= v-1$ ⇒ v是王且无入邻点 ⇒ v是唯一的王