弦图与区间图

Chordal Graph and Interval Graph

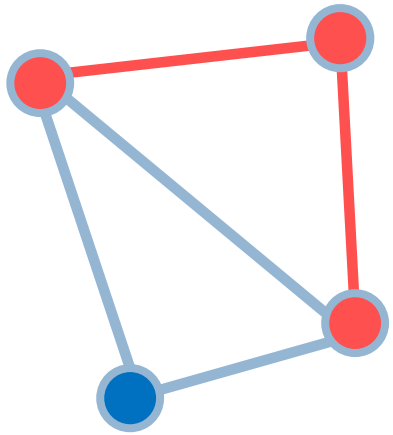
清华大学 陈丹琦

图的基本概念

- 子图 (subgraph)

  图 $G=(V,E)$

  $G'=(V',E'),V'\subseteq V,E'\subseteq E$ 为图G的子图



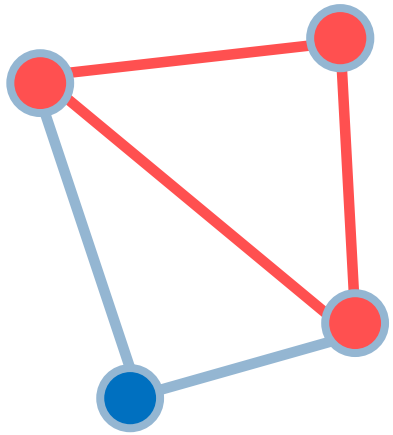
图的基本概念

- 诱导子图(induced subgraph)

  图 $G=(V,E)$

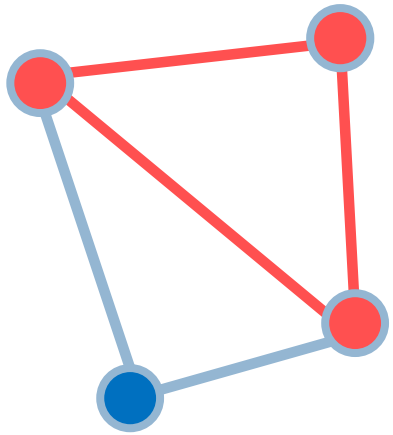
  $G'=(V',E'),V'\subseteq V,E'=\{(u,v)\,|\,u,v\in V',(u,v)\in E\}$

  称为图G的诱导子图



图的基本概念

- 团(clique)
  图G的一个子图$G'=(V',E')$，G'为关于V'的完全图。



图的基本概念

- 团(clique)
  图G的一个子图$G'=(V',E')$，G'为关于V'的完全图。
- 极大团(maximal clique)
  一个团是极大团当它不是其它团的子集。

图的基本概念

- 团(clique)
  图G的一个子图$G'=(V',E')$，G'为关于V'的完全图。
- 极大团(maximal clique)
  一个团是极大团当它不是其它团的子集。
- 最大团(maximum clique)
  点数最多的团。

$\omega(G)$　团数

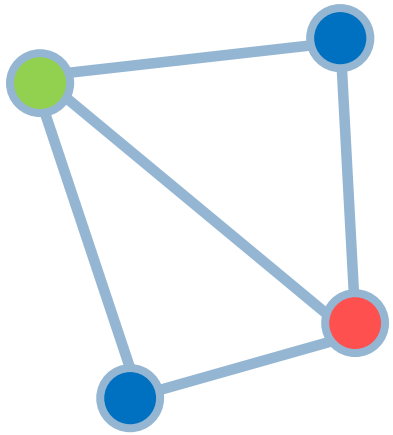
图的基本概念

- 最小染色(minimum coloring)

  用最少的颜色给点染色使相邻点颜色不同。

$\chi(G)$

色数

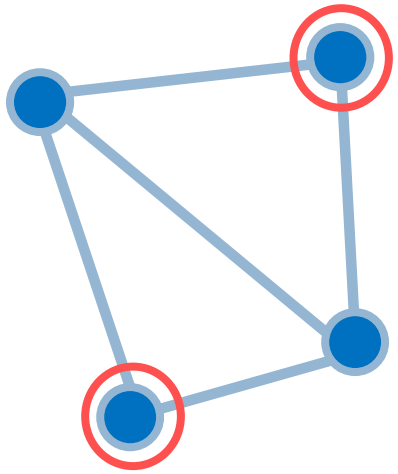


图的基本概念

- 最大独立集(maximum independent set)

  最大的一个点的子集使任何两个点不相邻。

$\alpha(G)$

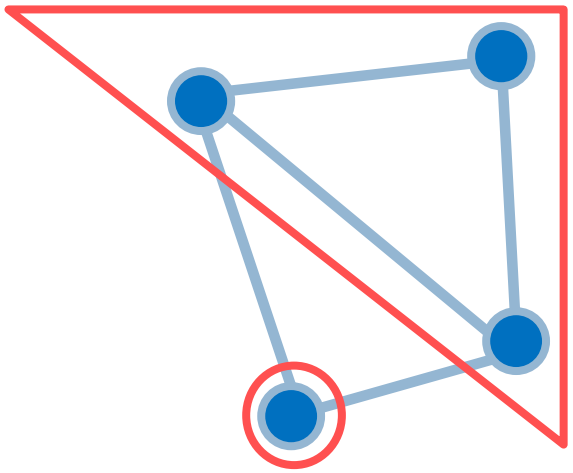


图的基本概念

- 最小团覆盖(minimum clique cover)

  用最少个数的团覆盖所有的点。

$\kappa(G)$



图的基本概念

$$\omega(G) \le \chi(G)$$

- 团数 ≤ 色数

图的基本概念

$$\omega(G) \le \chi(G)$$

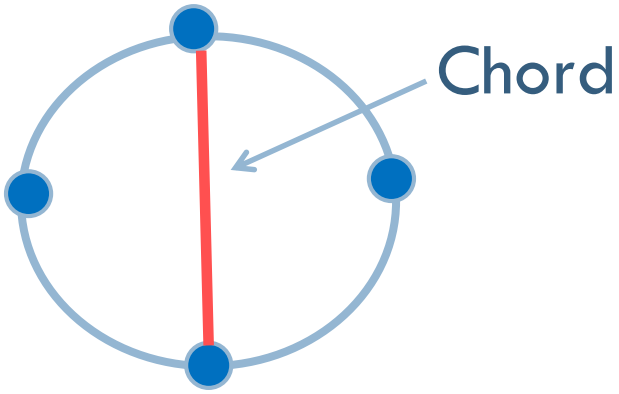
- 团数 ≤ 色数

$$\alpha(G) \le \kappa(G)$$

- 最大独立集数 ≤ 最小团覆盖数

弦图的概念

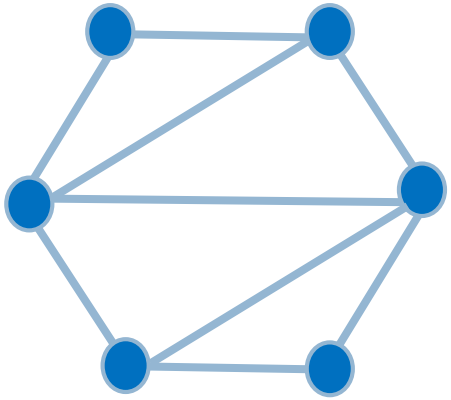
- 弦(chord)：连接环中不相邻的两个点的边。

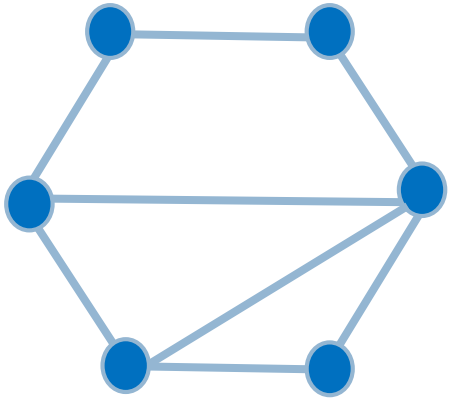


弦图的概念

- 弦(chord)：连接环中不相邻的两个点的边。
- 弦图(chordal graph)：一个无向图称为弦图

  当图中任意长度大于3的环都至少有一个弦。





弦图的概念

- 弦(chord)：连接环中不相邻的两个点的边。
- 弦图(chordal graph)：一个无向图称为弦图

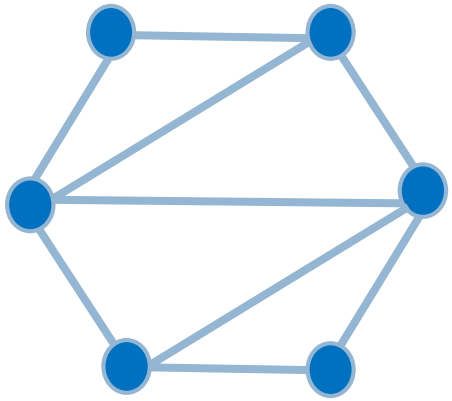
  当图中任意长度大于3的环都至少有一个弦。
- 弦图的每一个诱导子图一定是弦图。
- 弦图的任一个诱导子图不同构于C_n $(n > 3)$

弦图的判定

[例题] Zju1015 Fishing net

给定一个无向图，判定它是否为弦图。

- 单纯点(simplicial vertex)：设$N(v)$表示与点v相邻的点集。一个点称为单纯点当$\{v\} + N(v)$的诱导子图为一个团。

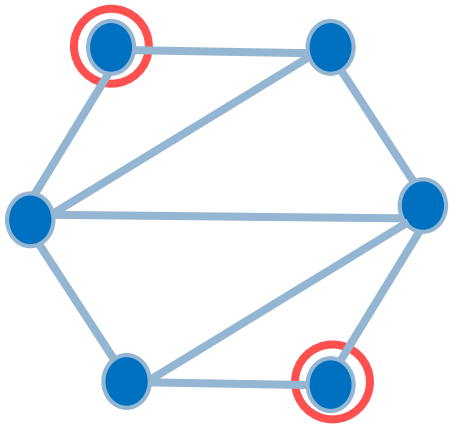


弦图的判定

[例题] Zju1015 Fishing net

给定一个无向图，判定它是否为弦图。

- 单纯点(simplicial vertex)：设$N(v)$表示与点v相邻的点集。一个点称为单纯点当$\{v\} + N(v)$的诱导子图为一个团。



弦图的判定

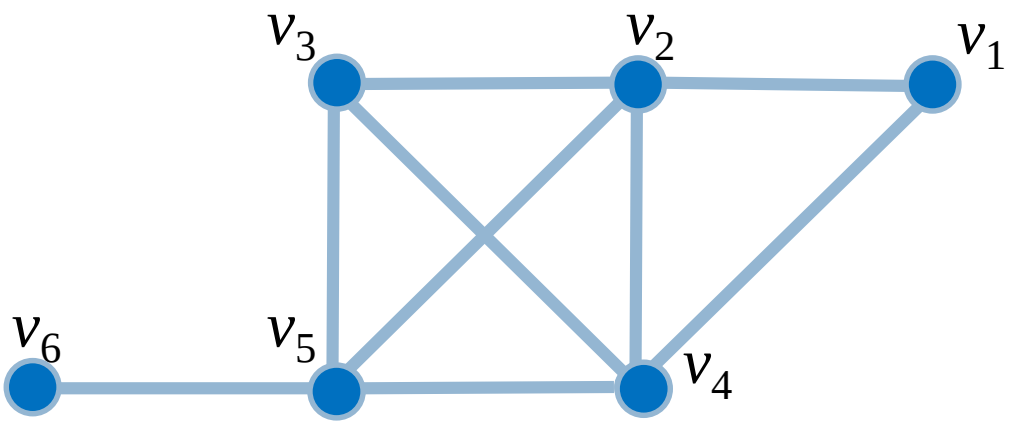
[例题] Zju1015 Fishing net

给定一个无向图，判定它是否为弦图。

- 单纯点(simplicial vertex)：设$N(v)$表示与点v相邻的点集。一个点称为单纯点当$\{v\} + N(v)$的诱导子图为一个团。
- 引理：任何一个弦图都至少有一个单纯点，不是完全图的弦图至少有两个不相邻的单纯点。

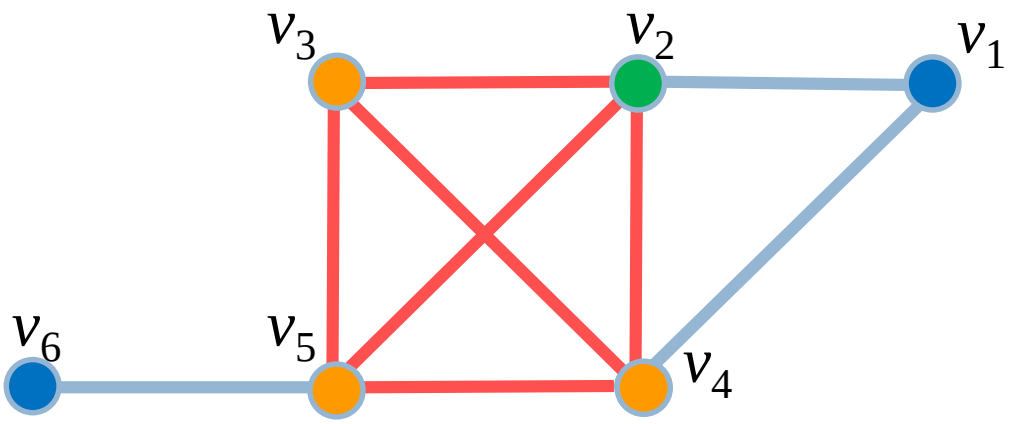
弦图的判定

- 完美消除序列(perfect elimination ordering)
- 　　定义：一个点的序列(每个点出现且恰好出现一次)$v_1, v_2, \ldots, v_n$满足v_i在$\{v_i, v_{i+1},\ldots,v_n\}$的诱导子图中为一个单纯点。



弦图的判定

- 完美消除序列(perfect elimination ordering)
- 　　定义：一个点的序列(每个点出现且恰好出现一次)$v_1, v_2, \ldots, v_n$满足v_i在$\{v_i, v_{i+1},\ldots,v_n\}$的诱导子图中为一个单纯点。



弦图的判定

- **定理：** 一个无向图是弦图当且仅当它有一个完美消除序列。

弦图的判定

- **定理：** 一个无向图是弦图当且仅当它有一个完美消除序列。
- **证明：** **充分性** 由引理知任何一个弦图都至少有一个单纯点以及弦图的诱导子图都是弦图。可以使用数学归纳法假设当点数$<n$的弦图一定有完美消除序列，那么点数为n的弦图的完美消除序列可以由一个单纯点加上剩余点的诱导子图的完美消除序列得到。

弦图的判定

- **定理：** 一个无向图是弦图当且仅当它有一个完美消除序列。
- **证明：** 必要性 反证若无向图存在一个长度 > 3的无弦环，不妨设环中在完美消除序列中出现在最前面的点为v，设环中v与v_1, v_2相连，根据完美消除序列的性质知v_1, v_2相连，与环无弦矛盾。所以无向图为弦图。

求完美消除序列

- 最朴素的算法：
- 每次找一个单纯点v，加入到完美消除序列中。
- 将v以及相关的边从图中删掉。
- 重复以上过程直到所有点都被删除(图为弦图，得到了完美序列)或不存在单纯点v(图不是弦图)。

求完美消除序列

- 最朴素的算法：
- 每次找一个单纯点v，加入到完美消除序列中。
- 将v以及相关的边从图中删掉。
- 重复以上过程直到所有点都被删除(图为弦图，得到了完美序列)或不存在单纯点v(图不是弦图)。

时间复杂度??　$O(n^4)$

求完美消除序列

- 最朴素的算法：
- 每次找一个单纯点v，加入到完美消除序列中。
- 将v以及相关的边从图中删掉。
- 重复以上过程直到所有点都被删除(图为弦图，得到了完美序列)或不存在单纯点v(图不是弦图)。

时间复杂度?? $O(n^4)$

下面介绍两个求完美消除序列$O(m+n)$的算法。

LexBFS算法

- 字典序广度优先搜索(Lexicographic BFS)
- 从n到1的顺序依次给点标号。
- 每个点维护一个list记录与它相邻的已标号点的标号，list中的标号按照按从大到小排序。
- 每次选择list字典序最大的未标号点标号。

1. For all vertices v, set $L(v) = \emptyset$;
2. For $i = n \ldots 1$
3. among all vertices $\neq v_{i+1}, \ldots, v_n$
4. pick up v_i with the lexicographically largest label $\mathrm{L}(v_i)$;
5. for each unnumbered vertex w that is adjacent to v
6. Set $L(w) = L(w) \circ i$

LexBFS算法

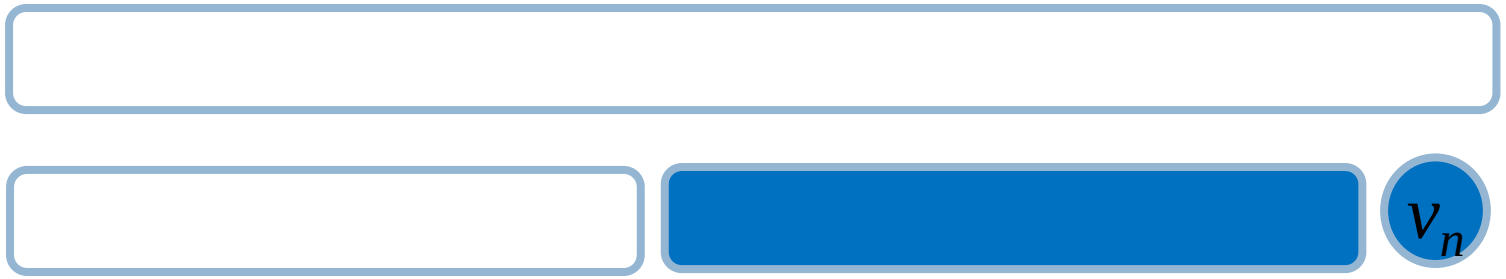
LexBFS与BFS不同在于每次扩展的节点加了特殊的顺序。

LexBFS算法

LexBFS与BFS不同在于每次扩展的节点加了特殊的顺序。

LexBFS算法

LexBFS与BFS不同在于每次扩展的节点加了特殊的顺序。

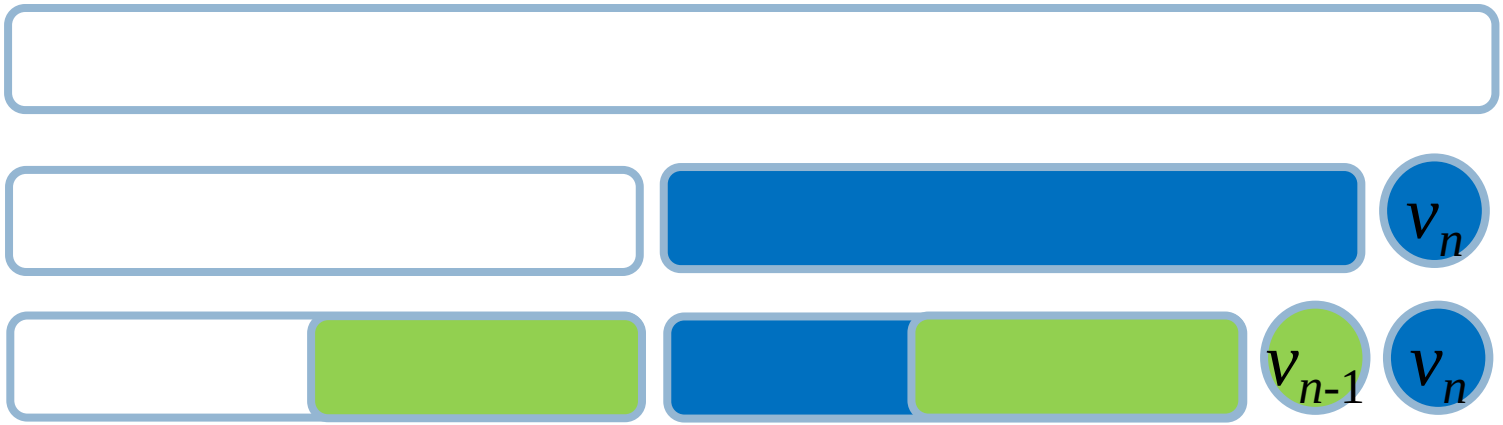


LexBFS算法

LexBFS与BFS不同在于每次扩展的节点加了特殊的顺序。

LexBFS算法

LexBFS与BFS不同在于每次扩展的节点加了特殊的顺序。

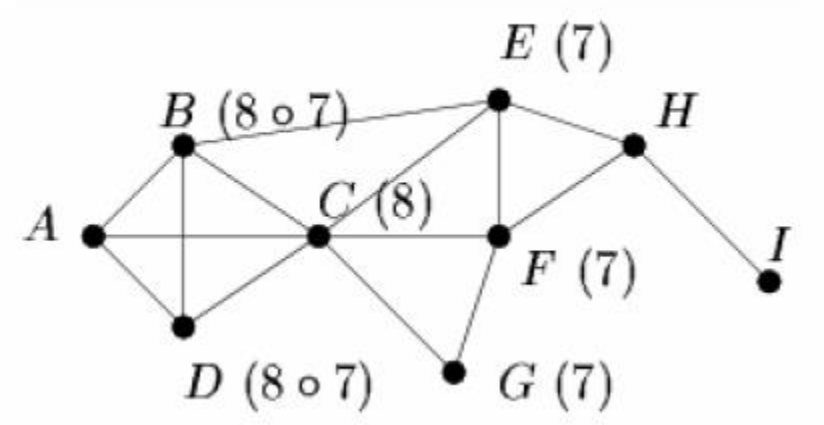


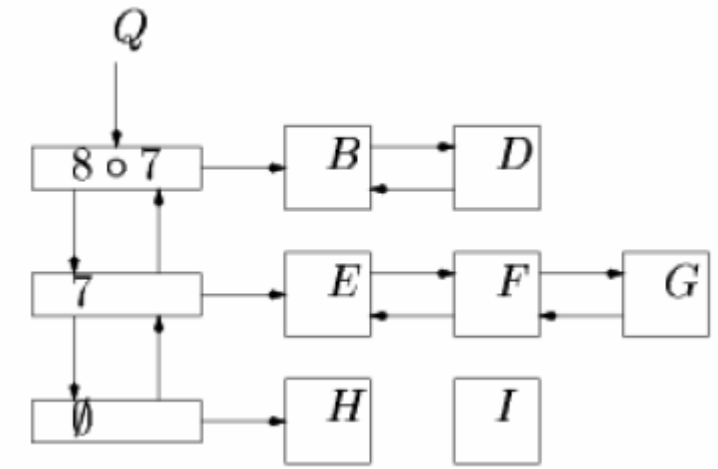
LexBFS算法

- 算法实现？

LexBFS算法

- 算法实现



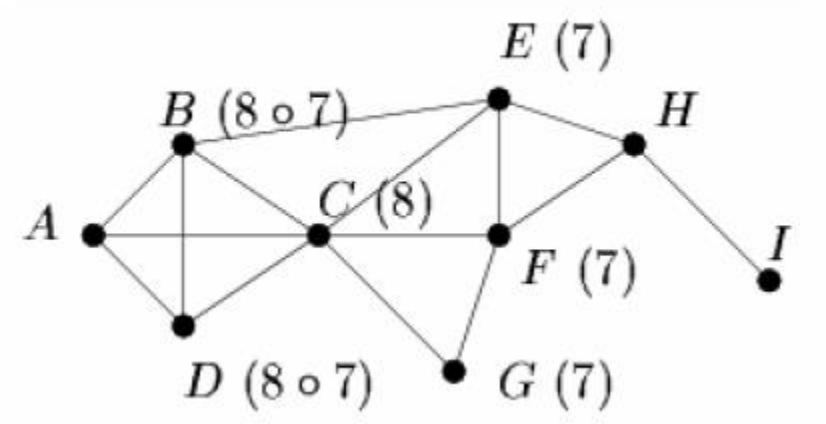


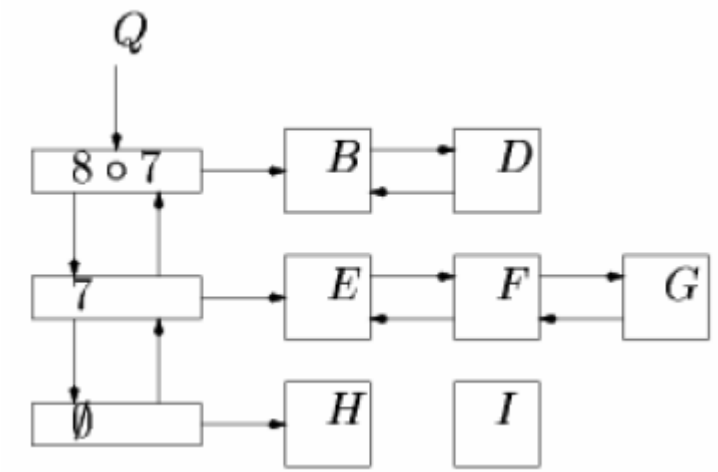
更新一个点的list最多新建一个桶。
任何时候桶的数目不超过n。

LexBFS算法

- 算法实现

O(m + n)!!!

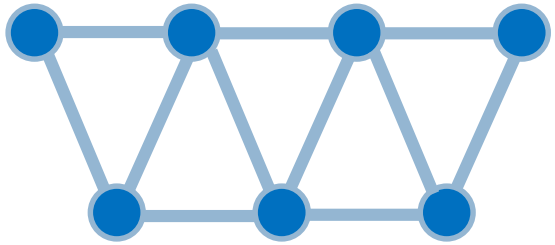




更新一个点的list最多新建一个桶。
任何时候桶的数目不超过n。

MCS算法

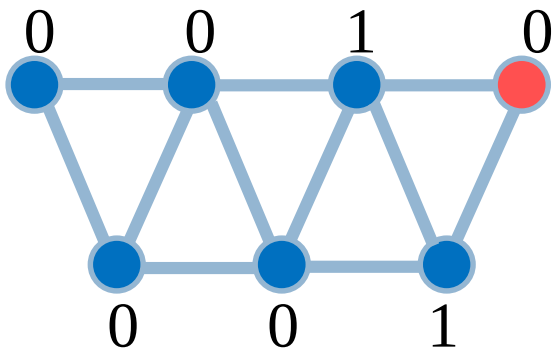
- 最大势算法 **Maximum Cardinality Search**
- 从*n*到1的顺序依次给点标号(标号为*i*的点出现在完美消除序列的第*i*个)。
- 设*label*[*i*]表示第*i*个点与多少个已标号的点相邻，每次选择*label*[*i*]最大的未标号的点进行标号。



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

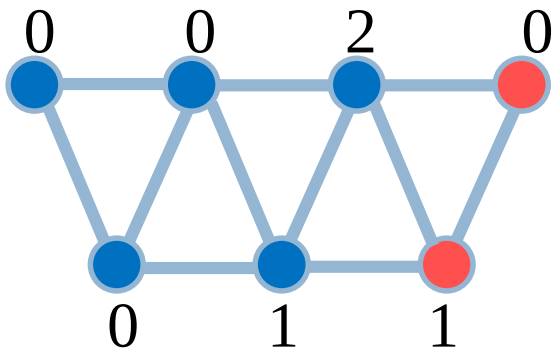
$i = 7$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

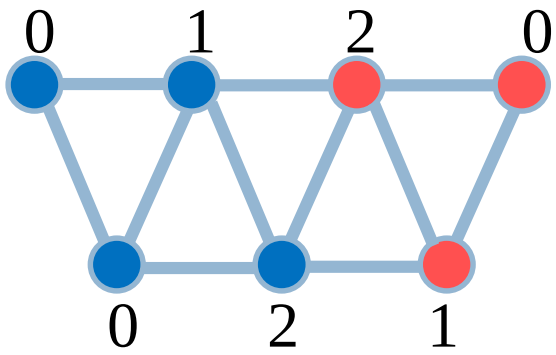
$i = 6$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

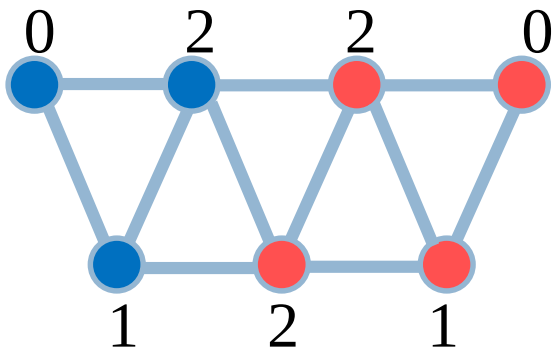
$i = 5$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

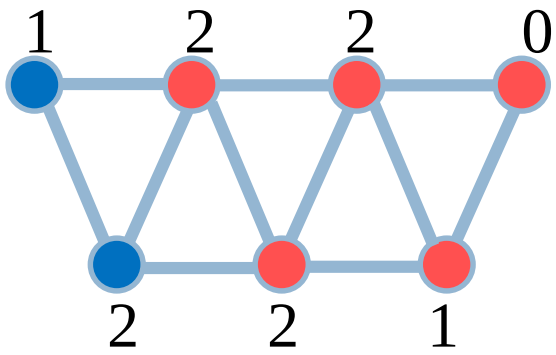
$i = 4$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

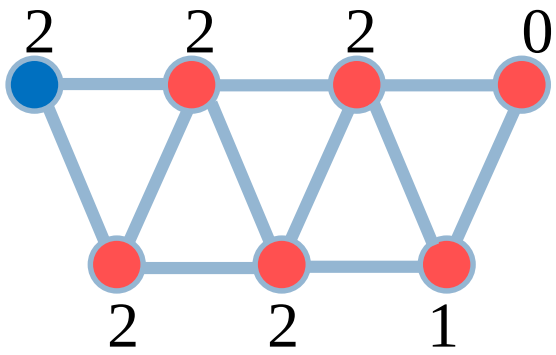
$i = 3$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

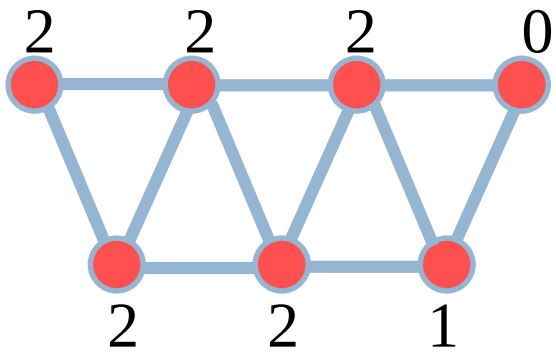
$i = 2$



MCS算法

- **最大势算法 Maximum Cardinality Search**
- 从n到1的顺序依次给点标号(标号为i的点出现在完美消除序列的第i个)。
- 设$label[i]$表示第i个点与多少个已标号的点相邻，每次选择$label[i]$最大的未标号的点进行标号。

$i = 1$



MCS算法

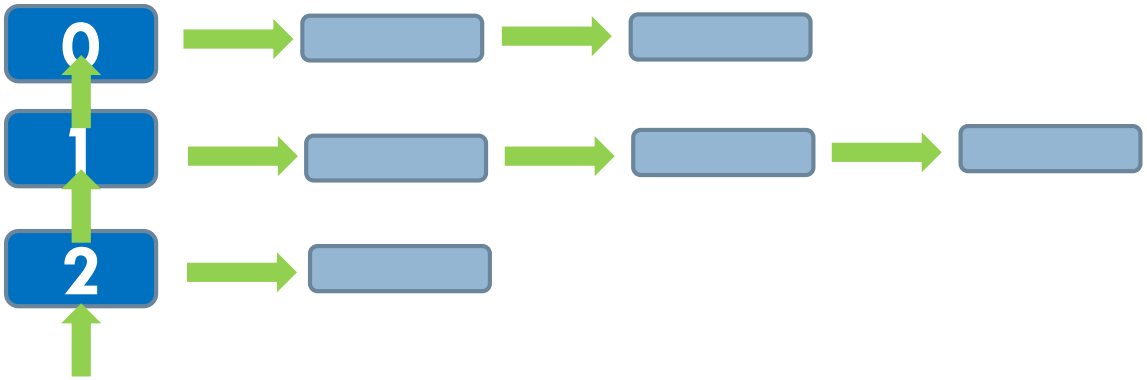
```
1:  procedure maximum cardinality search (G, σ)
2:      for all vertices v of G do
3:          set label[v] to zero
4:      end for
5:      for all i from n downto 1 do
6:          choose an unnumbered vertex v with largest label
7:          set σ(v) to i {number vertex v}
8:          for all unnumbered vertices w adjacent to vertex v do
9:              increment label[w] by one
10:         end for
11:     end for
12: end procedure
```

MCS算法

- 算法实现？

MCS算法

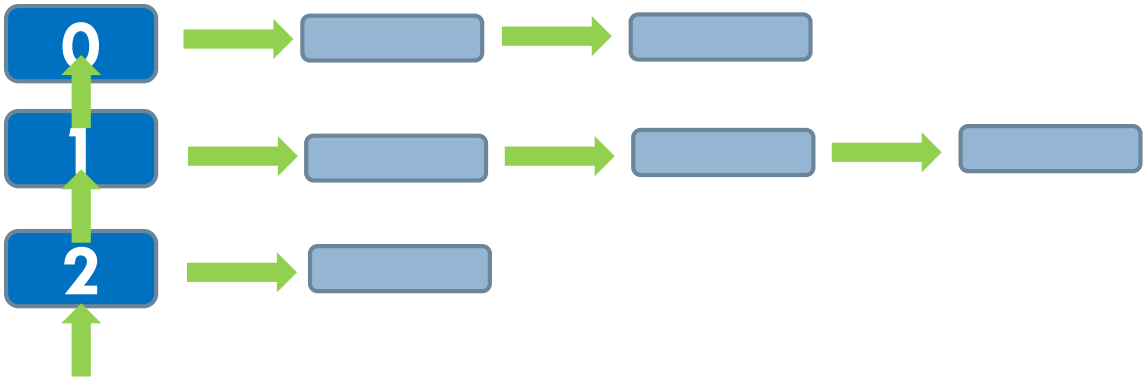
- 算法实现



MCS算法

- 算法实现

O(m + n)!!!



弦图的判定

- 判断一个序列是否为完美消除序列

弦图的判定

- 判断一个序列是否为完美消除序列
- 朴素的算法
- 依次判断 $\{v_{i+1},\ldots,v_n\}$ 中所有与 v_i 相邻的点是否构成了一个团。
- 时间复杂度：$O(\sum(\deg(v))^2)=O(mn)$

弦图的判定

- 判断一个序列是否为完美消除序列
- 优化后的算法
  - 设$\{v_{i+1},\ldots,v_n\}$中所有与v_i相邻的点依次为$v_{j1}, \ldots, v_{jk}$。
  - 只需判断v_{j1}是否与$v_{j2}, \ldots, v_{jk}$相邻即可。
  - 时间复杂度：$O(m+n)$

弦图的判定

- 判断一个序列是否为完美消除序列
- 优化后的算法
- 设$\{v_{i+1},\ldots,v_n\}$中所有与v_i相邻的点依次为$v_{j1}, \ldots, v_{jk}$。
- 只需判断v_{j1}是否与$v_{j2}, \ldots, v_{jk}$相邻即可。
- 时间复杂度: $O(m + n)$
- 弦图判定问题可以在$O(m + n)$的时间内解决。

弦图的极大团

- 设第i个点在弦图的完美消除序列第$p(i)$个。
- 令$N(v) = \{w \mid w$与v相邻且$p(w) > p(v)\}$
- 弦图的极大团一定是 $v \cup N(v)$ 的形式。
- 证明：

设点集V的诱导子图为弦图的极大团，设v为V中$p(i)$值最小的点即出现在完美消除序列中最前面的点。由于 $V \subseteq v \cup N(v)$ 为一个团，V为极大团所以 $V = v \cup N(v)$ 。

弦图的极大团

- 设第i个点在弦图的完美消除序列第$p(i)$个。
- 令$N(v) = \{w \mid w$与v相邻且$p(w) > p(v)\}$
- 弦图的极大团一定是 $v \cup N(v)$ 的形式。

- 推论：弦图最多有n个极大团。
- 如何找到弦图的所有极大团呢?
- 即判断每个 $v \cup N(v)$ 是否为极大团

弦图的极大团

- 判断 $v \cup N(v)$ 是否为极大团
- 设$A = v \cup N(v)$，若存在$B = w \cup N(w)$ 使得 $A \subseteq B$ 则A不是极大团。

弦图的极大团

- 判断 $v \cup N(v)$ 是否为极大团
- 设$A = v \cup N(v)$，若存在$B = w \cup N(w)$ 使得 $A \subseteq B$ 则A不是极大团。
- $p(w) < p(v)$

弦图的极大团

- 判断 $v \cup N(v)$ 是否为极大团
- 设$A = v \cup N(v)$，若存在$B = w \cup N(w)$ 使得 $A \subseteq B$ 则A不是极大团。
- $p(w) < p(v)$
- 设$next(v)$ 表示$N(v)$中最前的点。令w^*表示所有满足$A \subseteq B$的w中最后的一个点。

弦图的极大团

- 判断 $v \cup N(v)$ 是否为极大团
- 设$A = v \cup N(v)$，若存在$B = w \cup N(w)$ 使得 $A \subseteq B$ 则A不是极大团。
- $p(w) < p(v)$
- 设$next(v)$ 表示$N(v)$中最前的点。令w^*表示所有满足$A \subseteq B$的w中最后的一个点。
- $next(w^*) = v$ (否则$next(w^*)$也是满足条件的w)

弦图的极大团

- $Next(w) = v$
- $v \cup N(v) \subseteq w \cup N(w)$ 当且仅当

  $|N(v)| + 1 \leq |N(w)|$

弦图的极大团

- $Next(w) = v$
- $v \cup N(v) \subseteq w \cup N(w)$ 当且仅当 $|N(v)| + 1 \leq |N(w)|$
- 只需判断是否存在一个w，满足$Next(w) = v$且$|N(v)| + 1 \leq |N(w)|$即可。

  时间复杂度：$O(m + n)$

弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》

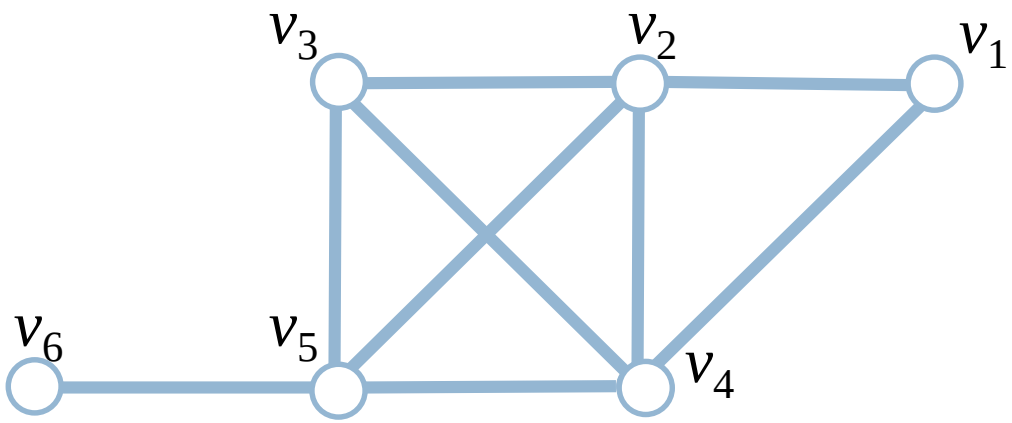
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》

- 完美消除序列从后往前依次给每个点染色，给每个点染上可以染的最小的颜色。

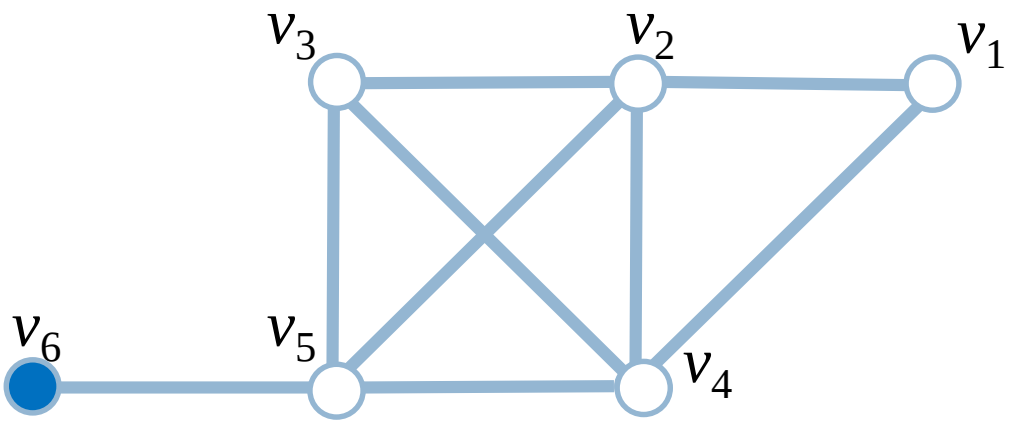
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



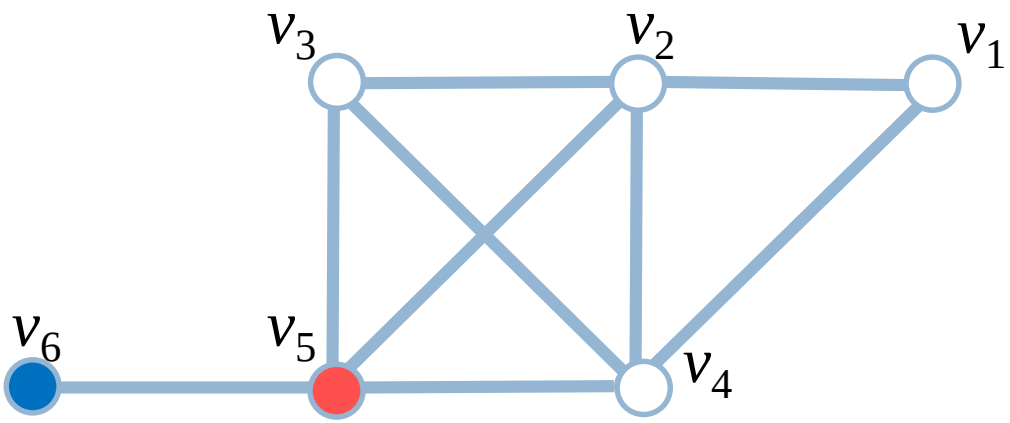
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



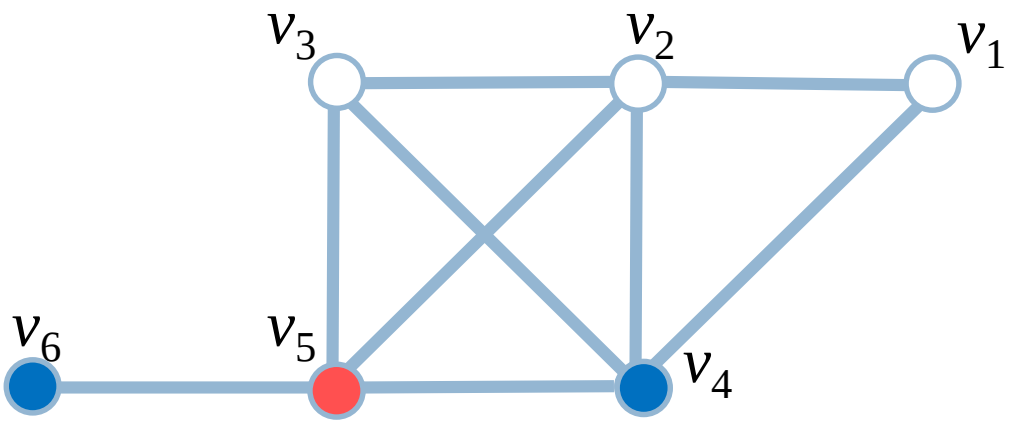
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



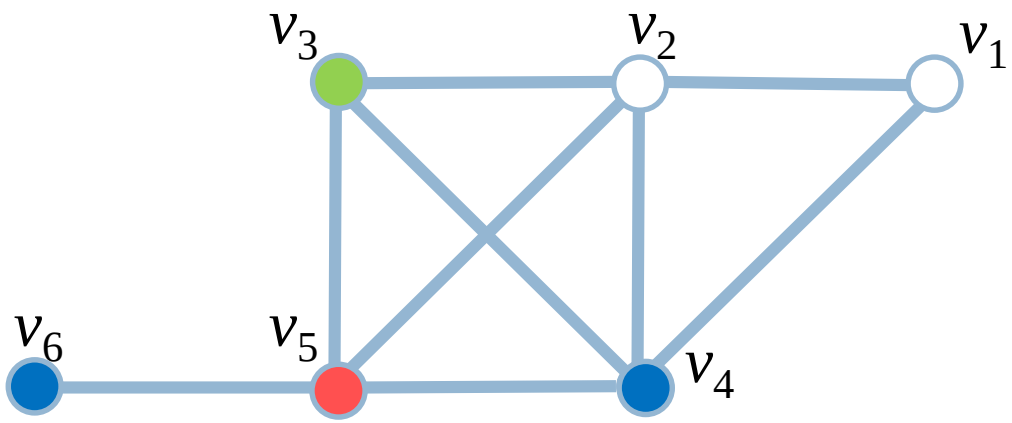
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



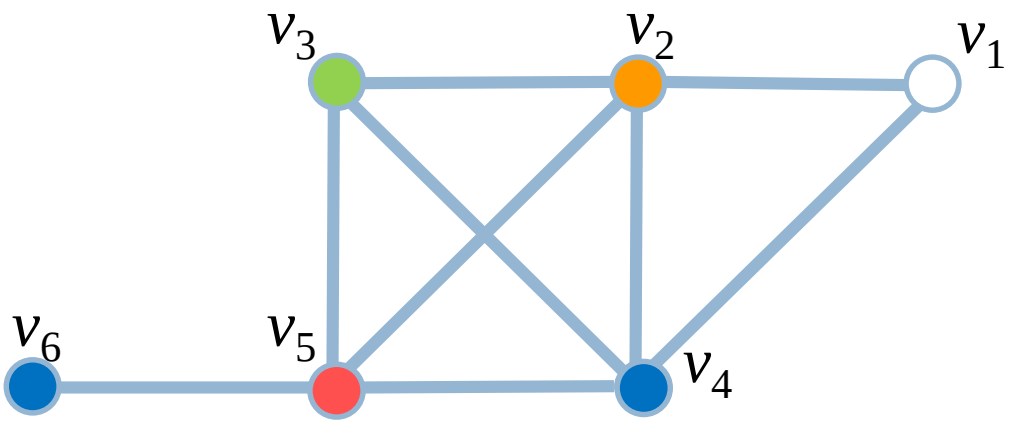
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



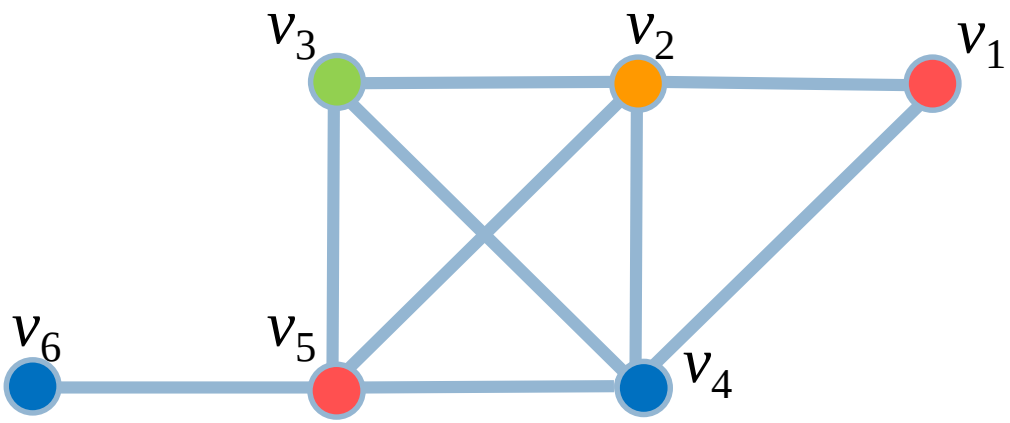
弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》



弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》
- 证明：
- 设使用了T种颜色，则$T \geq$色数
- $T =$ 团数 $\leq$ 色数
- 团数 = 色数 = T
- 时间复杂度： $O(m+n)$

弦图的点染色问题

- 用最少的颜色给每个点染色使得相邻的点染的颜色不同。
- [例题] HNOI2008《神奇的国度》
- 证明：
- 设使用了T种颜色，则$T \geq$色数
- $T =$ 团数 $\leq$ 色数
- 团数 = 色数 = T
- 时间复杂度： $O(m+n)$

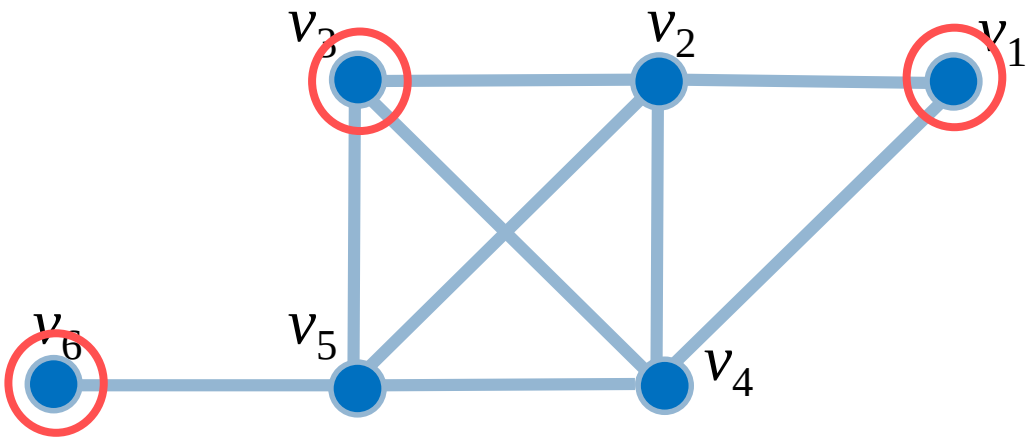
团数 = 色数!!!

弦图的最大独立集和最小团覆盖

- 最大独立集
- 选择最多的点使得任意两个点不相邻。

弦图的最大独立集和最小团覆盖

- 最大独立集
- 　　选择最多的点使得任意两个点不相邻。
- **Sol** 完美消除序列从前往后能选就选。

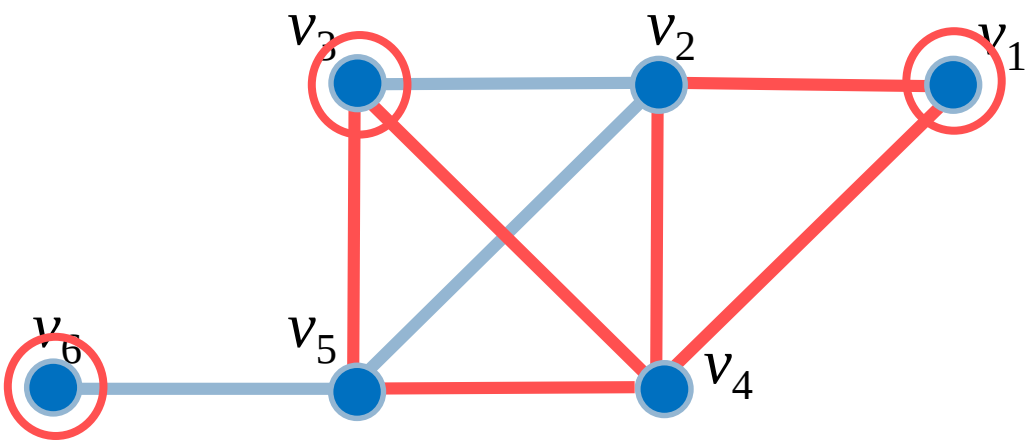


弦图的最大独立集和最小团覆盖

- 最小团覆盖
- 　用最少个数的团覆盖所有的点。

弦图的最大独立集和最小团覆盖

- 最小团覆盖
- 　　用最少个数的团覆盖所有的点。
- **Sol**　设最大独立集为$\{p_1, p_2, \ldots, p_t\}$，则$\{p_1 \cup N(p_1), \ldots, p_t \cup N(p_t)\}$为最小团覆盖。



弦图的最大独立集和最小团覆盖

- 证明
  - $\{p_1, p_2, \ldots, p_t\}$为一个独立集。

$$t \leq \alpha(G)$$

弦图的最大独立集和最小团覆盖

- 证明
  - $\{p_1, p_2, \ldots, p_t\}$为一个独立集。

    $t \leq \alpha(G)$
  - $\{p_1 \cup N(p_1), \ldots, p_t \cup N(p_t)\}$为一个团覆盖。

    $t \geq \kappa(G)$

弦图的最大独立集和最小团覆盖

- 证明
  - $\{p_1, p_2, \ldots, p_t\}$为一个独立集。

    $t \leq \alpha(G)$
  - $\{p_1 \cup N(p_1), \ldots, p_t \cup N(p_t)\}$为一个团覆盖。

    $t \geq \kappa(G)$
  - 由$\alpha(G) \leq \kappa(G)$知$\alpha(G) = \kappa(G) = t$
  - 最大独立集数 = 最小团覆盖数!!!

完美图与伴完美图

- 完美图(Perfect Graph)的概念
- 　　一个图G称为完美图若它的每一个诱导子图都满足 $\omega(G) = \chi(G)$。

完美图与伴完美图

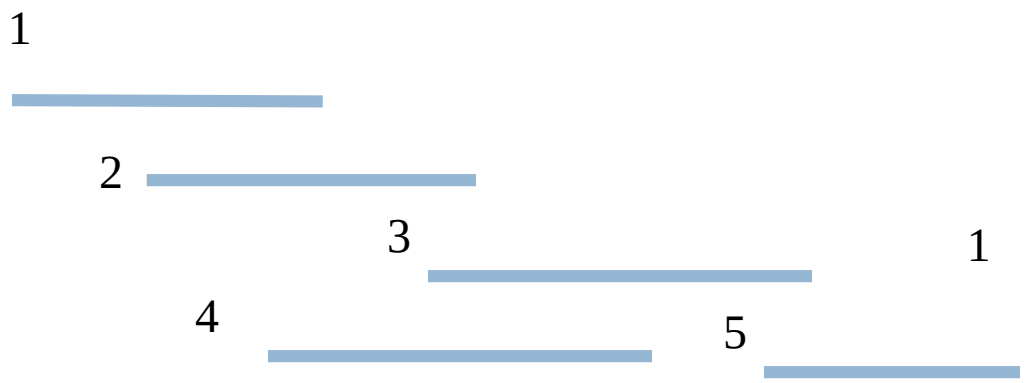
- 完美图(Perfect Graph)的概念
- 　　一个图G称为完美图若它的每一个诱导子图都满足 $\omega(G)=\chi(G)$。
- 伴完美图(Co-perfect Graph)的概念
- 　　一个图G称为完美图若它的每一个诱导子图都满足 $\alpha(G)=\kappa(G)$。

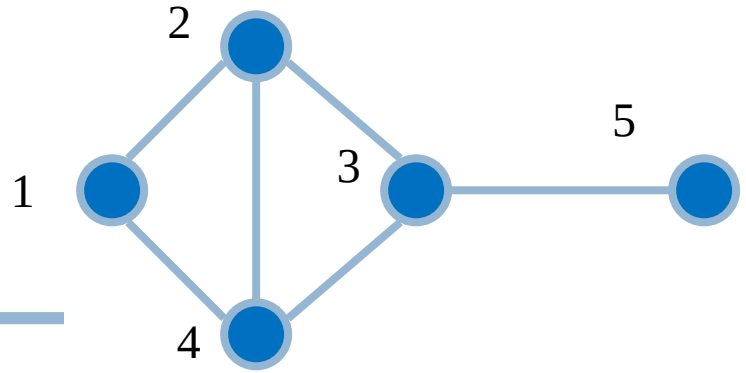
完美图与伴完美图

- 完美图(Perfect Graph)的概念
- 　　一个图G称为完美图若它的每一个诱导子图都满足 $\omega(G)=\chi(G)$。
- 伴完美图(Co-perfect Graph)的概念
- 　　一个图G称为完美图若它的每一个诱导子图都满足 $\alpha(G)=\kappa(G)$。
- 完美图 ＝伴完美图
- 弦图属于完美图。

区间图

- 区间图(Interval Graph)定义
- 给定一些区间，定义一个相交图为每个顶点表示一个区间，两个点有边当且仅当两个区间的交集非空。
- 一个图为区间图当它是若干个区间的相交图。



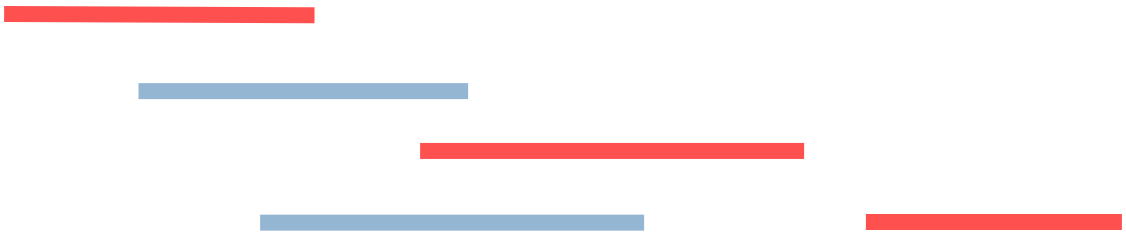


区间图

- 区间图一定是弦图。
- 证明：
- 若区间图中存在一个长度 >3的无弦环 $\{v_0, v_1, \ldots, v_{l-1}, v_l = v_0\}, l > 3$，设第$i$个点对应的区间为$I_i$。由$I_i$与$I_{i+1}$相交，取 $p_i \in I_i \cap I_{i+1}$，由于I_i与I_{i+2}不相交，则p_i一定严格递增或严格递减。由 $p_0 \in I_0$ 及 $p_{l-1} \in I_0$ 得到 $p_1 \in I_0$，与I_0与I_2不相交矛盾。所以区间图一定是弦图。

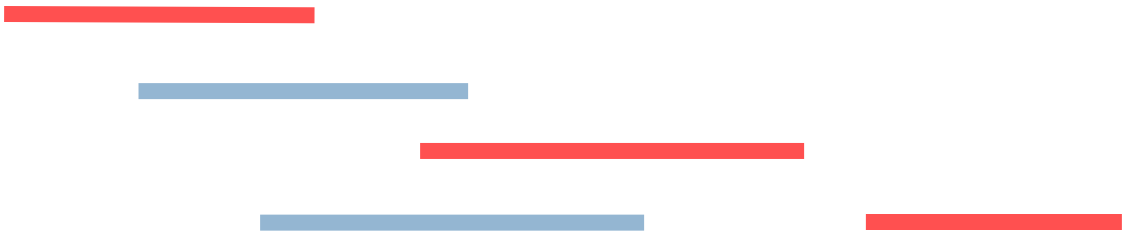
经典问题

- [例题1] 给定n个区间，要求选择最多的区间使得区间不互相重叠。



经典问题

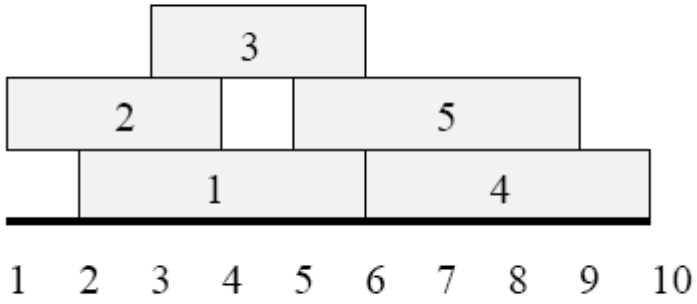
- [例题1] 给定n个区间，要求选择最多的区间使得区间不互相重叠。



区间图的最大独立集

经典问题

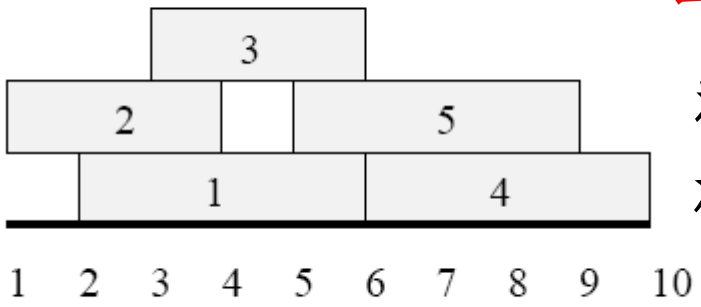
- [例题2] Tetris
- 有n个积木，高度均为1，第i个积木的宽度范围为$[L_i, R_i]$，选择一个积木的下落顺序使得最后积木总高度尽可能小。



经典问题

- [例题2] Tetris
- 有n个积木，高度均为1，第i个积木的宽度范围为$[L_i, R_i]$，选择一个积木的下落顺序使得最后积木总高度尽可能小。

区间图的最小染色



积木下落顺序：按照颜色标号从小到大落下。

区间图

- 给定n个区间，所对应的区间图为G
- G的一个完美消除序列：

  将所有的区间按照右端点从小到大排序。

区间图

- 给定n个区间，所对应的区间图为G
- G的一个完美消除序列：

  将所有的区间按照右端点从小到大排序。

- [例题1] 将所有的区间按照右端点从小到大排序，依次处理每个区间，如果与已选区间没有重叠则选择该区间。
- 时间复杂度：$O(n\log_2 n)$

区间图

- 给定n个区间，所对应的区间图为G
- G的一个完美消除序列：

  将所有的区间按照右端点从小到大排序。

- [例题2] 将所有的区间按照右端点从大到小排序，依次处理每个区间，选择一个可以染的最小的颜色染色。
- 线段树或堆维护
- 时间复杂度：$O(n\log_2 n)$

树的分解(Tree Decomposition)

- 定义
- 对于一个无向图$G = (V, E)$，树的分解定义为(X, T)，$X = \{X_1, X_2, \ldots, X_m\}$其中$X_i$为$V$的一个子集，$T$是一棵树，点集为$X$。$T$满足以下几个性质：

树的分解(Tree Decomposition)

- 定义
- 对于一个无向图$G = (V, E)$，树的分解定义为(X, T)，$X = \{X_1, X_2, \ldots, X_m\}$其中$X_i$为$V$的一个子集，$T$是一棵树，点集为$X$。$T$满足以下几个性质：
  - $X_1 \cup X_2 \cup \ldots \cup X_m = V$

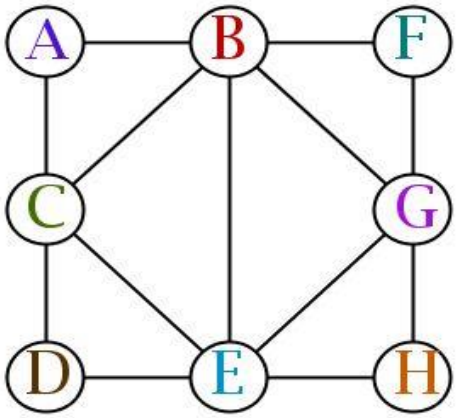
树的分解(Tree Decomposition)

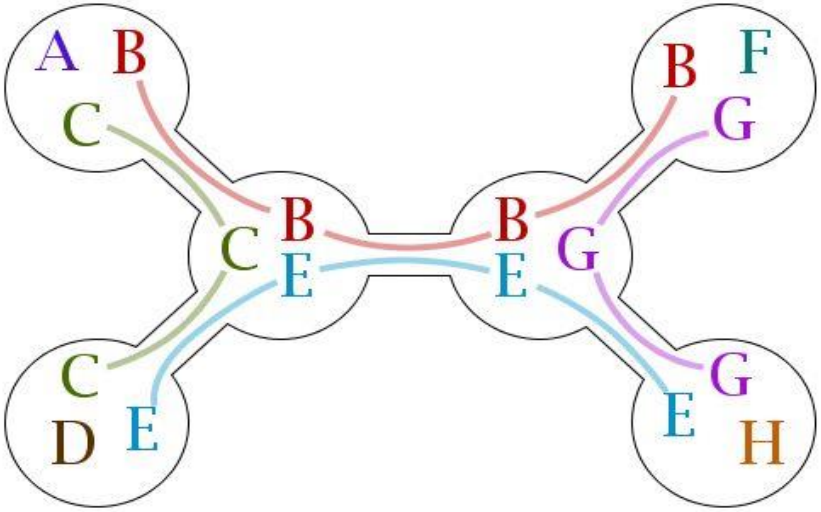
- 定义
- 对于一个无向图$G = (V, E)$，树的分解定义为(X, T)，$X = \{X_1, X_2, \ldots, X_m\}$其中$X_i$为$V$的一个子集，$T$是一棵树，点集为$X$。$T$满足以下几个性质：
  - $X_1 \cup X_2 \cup \ldots \cup X_m = V$
  - 图G中任何一条边(u, v)存在一个X_i使得$u, v \in X_i$。

树的分解(Tree Decomposition)

- 定义
- 对于一个无向图$G = (V, E)$，树的分解定义为(X, T)，$X = \{X_1, X_2, \ldots, X_m\}$其中$X_i$为$V$的一个子集，$T$是一棵树，点集为$X$。$T$满足以下几个性质：
  - $X_1 \cup X_2 \cup \ldots \cup X_m = V$
  - 图G中任何一条边(u, v)存在一个X_i使得$u, v \in X_i$。
  - 对于每一个点v，$P(v)=\{X_i \mid v \in X_i\}$，则$T$中$P(v)$的诱导子图是连通的。

树的分解(Tree Decomposition)



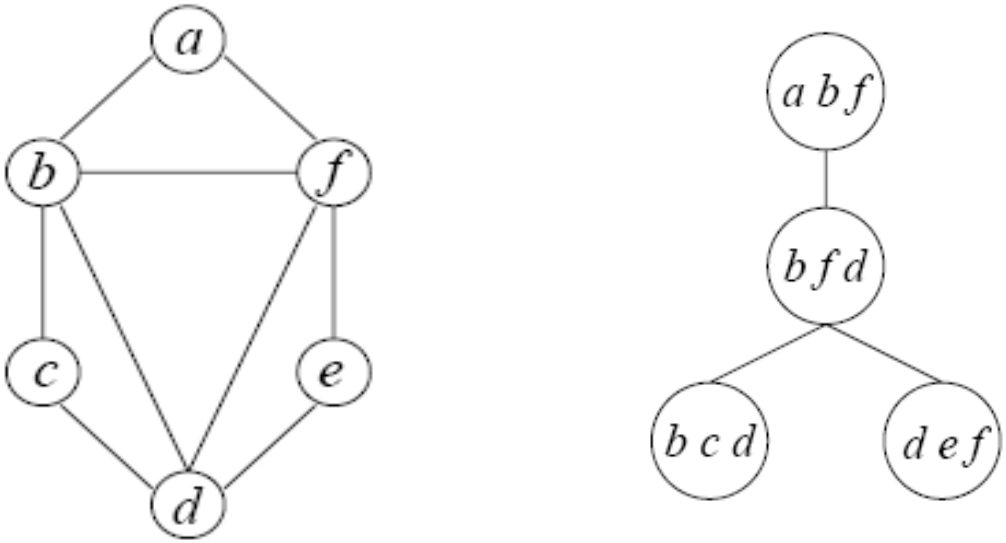


Clique Tree

- 一个无向图*G*的极大团树*T*(Clique Tree)定义为：

  *T*的顶点为图*G*的所有极大团

  包含每个点的所有极大团为*T*的一个连通子图。



Clique Tree

- Clique Tree是一种tree decomposition.

Clique Tree

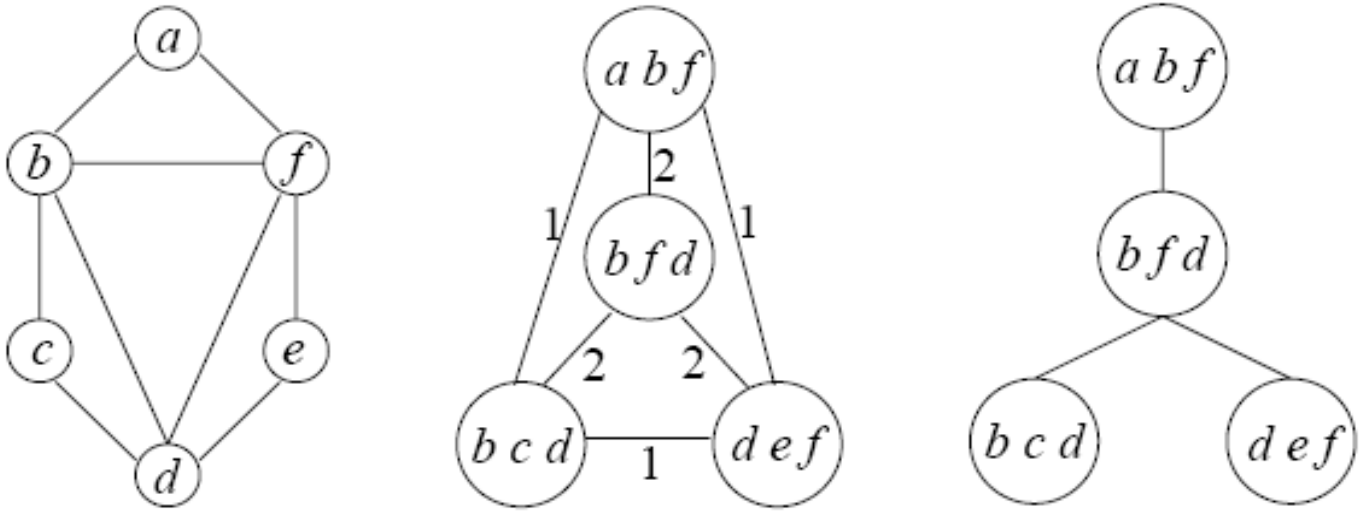
- Clique Tree是一种tree decomposition.
- 一个无向图***G***为弦图当且仅当它存在一个Clique Tree.

Clique Tree

- Clique Tree是一种tree decomposition.
- 一个无向图***G***为弦图当且仅当它存在一个Clique Tree.
- 构建弦图的一个Clique Tree
- 找出弦图的所有极大团。
- 构图 G'：极大团为点，两个点之间的边权为两个极大团的交集的点的个数。
- 求图 G'的一个最大生成树。

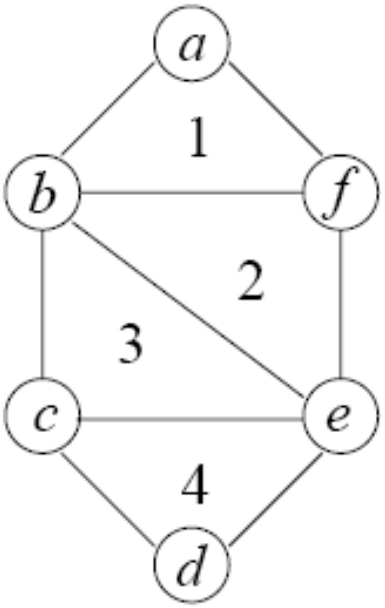
Clique Tree

- Clique Tree是一种tree decomposition.
- 一个无向图***G***为弦图当且仅当它存在一个Clique Tree.



Clique Tree

- 一个无向图*G*为区间图当且仅当它存在一个Clique Tree是一条链。

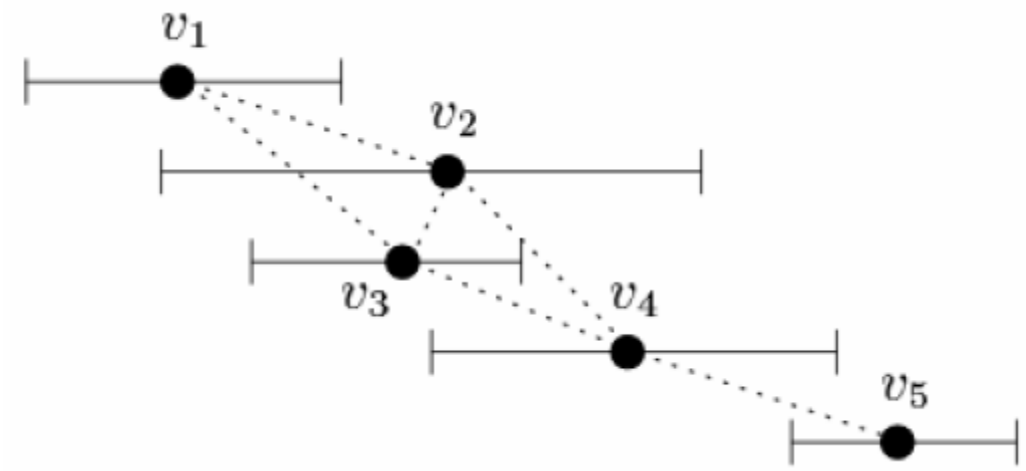


区间图的判定

- 判断是否是弦图 $O(m+n)$
- 如果是弦图，找出所有的极大团 $O(m+n)$
- 判断是否存在一个连续的极大团序列即包含每个点的极大团在序列中都是连续的。

区间图的判定

- 判断是否是弦图 $O(m+n)$
- 如果是弦图，找出所有的极大团 $O(m+n)$
- 判断是否存在一个连续的极大团序列即包含每个点的极大团在序列中都是连续的。



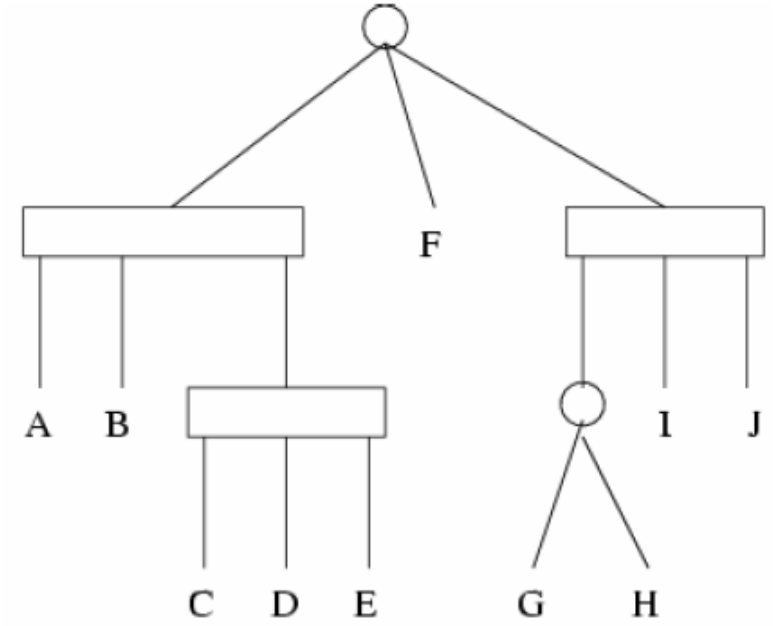
	C_1	C_2	C_3
v_1	1	0	0
v_2	1	1	0
v_3	1	1	0
v_4	0	1	1
v_5	0	0	1

区间图的判定

- **PQ树**
- 给定一个0, 1矩阵要求将矩阵的列重排使得每一行的1是连续的。

区间图的判定

- **PQ树**
- 给定一个0, 1矩阵要求将矩阵的列重排使得每一行的1是连续的。

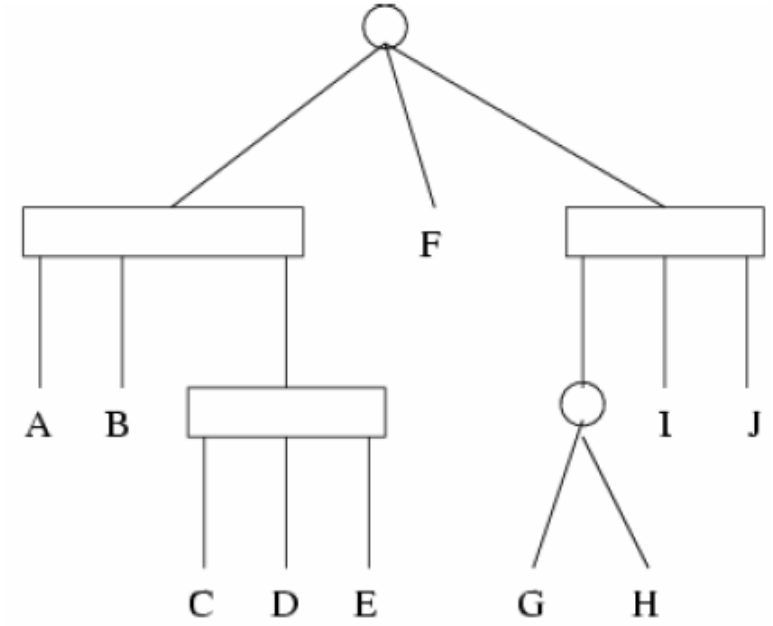


○ *P*节点
[子节点可以任意重排]

▭ *Q*节点
[原顺序或反序]

区间图的判定

- **PQ树**
- 给定一个0, 1矩阵要求将矩阵的列重排使得每一行的1是连续的。



○ *P*节点
[子节点可以任意重排]

▭ *Q*节点
[原顺序或反序]

FABEDCHGIJ

区间图的判定

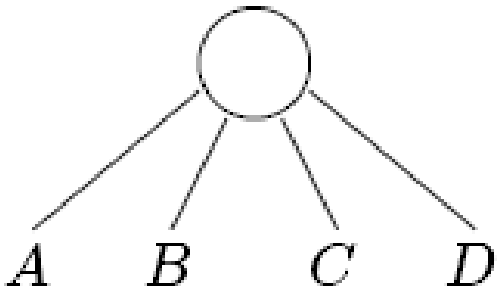
- 元素集合为X, 初始元素顺序任意。
- 有若干个限制, 每一个限制集合I表示要将I内的元素变成连续的。
- 利用PQ树每一个限制可以在$O(|I|)$内解决, 总的时间复杂度为 $O(|X|+\sum|I|)$。

区间图的判定

- 元素集合为X, 初始元素顺序任意。
- 有若干个限制, 每一个限制集合I表示要将I内的元素变成连续的。
- 利用PQ树每一个限制可以在$O(|I|)$内解决, 总的时间复杂度为 $O(|X|+\sum|I|)$。
- 区间图判定
- 最多有n个极大团, $|X| \le n$
- 每个点最多属于$deg(v)$个极大团, $\sum|I| \le 2m$
- 时间复杂度为$O(n+m)$。

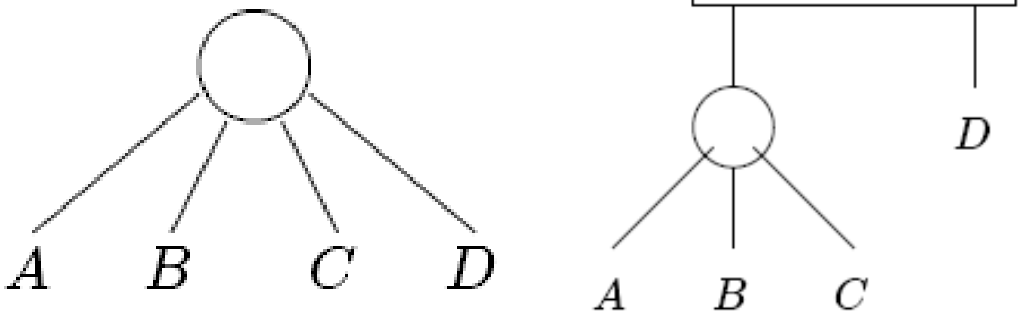
例

- $X = \{A, B, C, D\}$



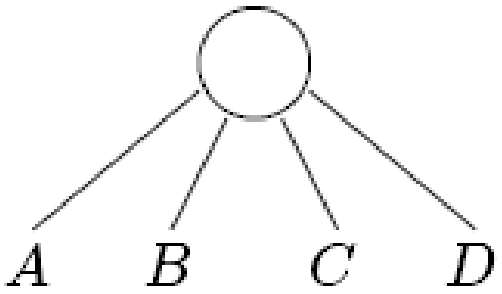
例

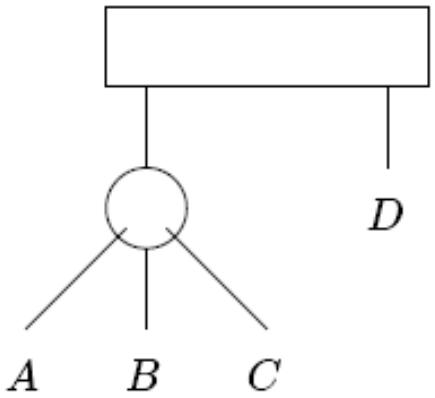
- $X = \{A, B, C, D\}$
- $I_1 = \{A, B, C\}$

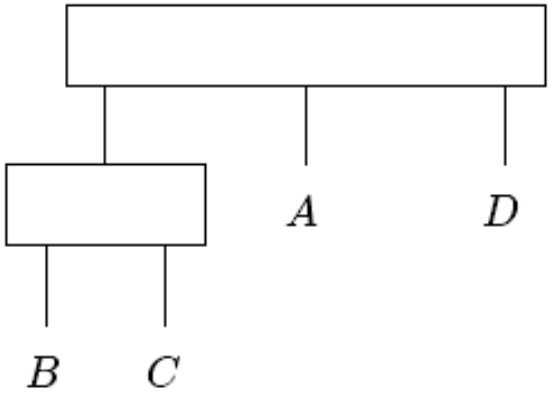


例

- $X = \{A, B, C, D\}$
- $I_1 = \{A, B, C\}$
- $I_2 = \{A, D\}$







区间图的判定

- PQ树的实现 很复杂 -_____-
- 感兴趣的同学欢迎与我联系☺。

区间图的判定

- **PQ树的实现** 很复杂 -_____-
- 感兴趣的同学欢迎与我联系☺。

- 其他区间图判定方法：

- Hsu [1992], Hsu and Ma [1999] decomposition, off-line
- 《A new Test for Interval Graphs》

Thank you.

My Email: cdq10131@gmail.com

Others

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。

那些收取财富值的要不得啊。怎么能这样呢。

难道你是作者？你有什么权利收取？

你传上来了，大家都很感谢你。但是为什么非要加钱在上面呢？

这样有什么意思呢？

还有的1财富值有的2财富值。

你们营销啊！

传个免费的。