Permutation Graphs

- Let π be a permutation on $N_n = \{1, 2, \dots, n\}$.

- Let us think of π as a sequence $[\pi_1, \pi_2, \dots, \pi_n]$ so, for example, the permutation $\pi = [4, 3, 6, 1, 5, 2]$ has $\pi_1 = 4$, $\pi_2 = 3$, ect.

- Notice that π_i^{-1} is the position in the sequence where the number i can be found; in our example $\pi_4^{-1} = 1$, $\pi_3^{-1} = 2$, etc

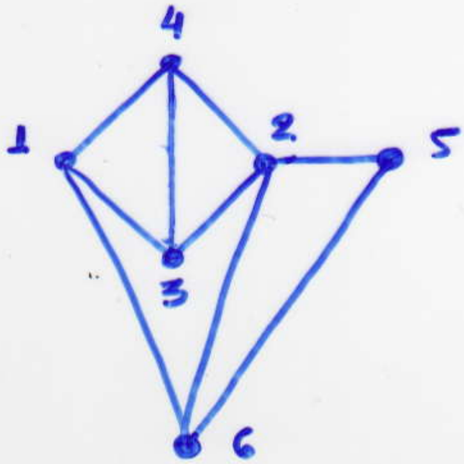
- We define the inversion Graph $G[\pi] = (V, E)$ of π as follows:

$$V = \{1, 2, \dots, n\}$$

$$(i, j) \in E \iff (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0$$

- In our example, both 4 and 3 are connected to 1, whereas neither 5 nor 2 is connected to 1.

- Definition: An undirected graph G is called a permutation graph if there exists a permutation π on N_n such that $G \cong G[\pi]$.



The graph $G[4,3,6,1,5,2]$

- Notice what happens when we reverse the sequence π.
  - the permutation graph we obtain is the complement of $G[\pi]$; $\pi^r = [2,5,1,6,3,4]$
  - $G[\pi^r] = \overline{G[\pi]}$
- This shows that the complement of a permutation graph is also a permutation graph.

- **Theorem**: An undirected graph G is a permutation graph iff G and $\bar{G}$ are comparability graphs.

- Let (V, F_1) and (V, F_2) be transitive orientations of $G=(V,E)$ and $\bar{G}=(V,\bar{E})$, respectively.

- Then, (i) $(V, F_1 + F_2)$ is acyclic.

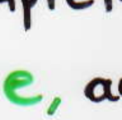
  (ii) $(V, F_1^{-1} + F_2)$ is acyclic.

Pls, prove (i) and (ii).

Construction of a permutation π : $G \cong G[\pi]$.

- An acyclic orientation of a complete graph K_n on n vertices is transitive, and it determines a unique linear order of the vertices.

- show: Golumbic p. 42



$v_n \equiv$ in-degree $n-1$

- Let F be an orientation of K_n. We call F a transitive orientation if, for all triples of vertices,

$$xy \in F,\ yz \in F \Rightarrow xz \in F$$

- Theorem: Let F be an orientation of K_n. The following statements are equivalent.

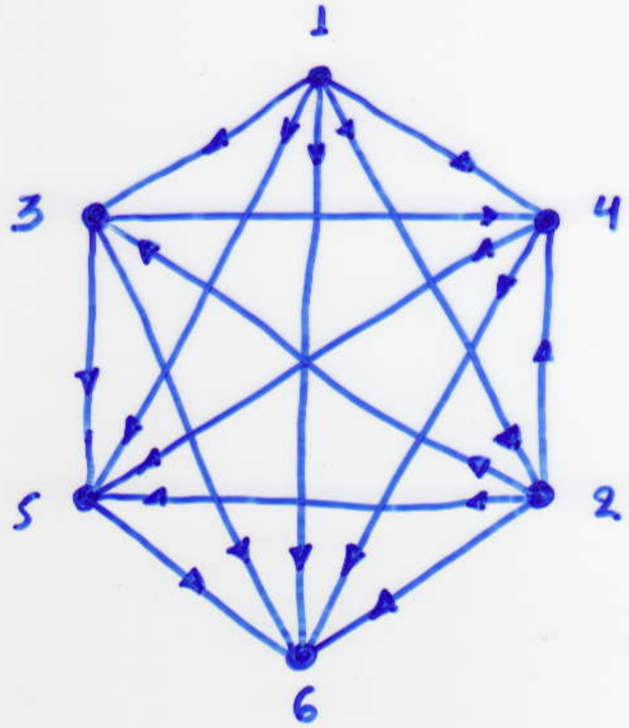
  (i) F is a transitive orientation.
  (ii) F is acyclic.

  Moreover, the vertices can be linearly ordered $[v_1, v_2, \ldots, v_n]$ such that

  (iii) v_i has in-degree $i-1$ in F, for all i, and
  (iv) $v_i v_j \in F$ iff $i<j$.

- This linear ordering of the vertices is unique.

- The following figure shows a transitive orientation and the linear ordering of its vertices.



- This theorem provides us with a $O(n)$ time algorithm for recognizing transitive orientations.

  - First, caluutate the in-degree of each vertex;
  - then, using a Boolean vector, verify that there are no duplicates among the in-degrees.

- A slightly more general problem than recognizing transitive orientations is that of topological sorting an arbitrary acyclic oriented graph $G=(V,F)$.

- What we seek is a linear ordering of the vertices $[v_1, v_2, \ldots, v_n]$ which is consistent with the edges of G; that is

$$v_i v_j \in F \Rightarrow i<j \quad (\text{for all } i,j).$$

  - such an ordering is called a topological sorting of G.

- One method for finding such an ordering is the following:

```
for i ← n to 1 step -1 do
    • locate a sink v of the remaining graph; v ≡ v_i;
    • delete v and all edges incident on v from G;
end;
```

- End-show;

- Consider the following procedure which constructs a permutation π such that $G \cong G[\pi]$.

- Recall: (V, F_1), (V, F_2) are the transitive orientations of G and $\bar{G}$, respectively.

- <u>Procedure</u>:

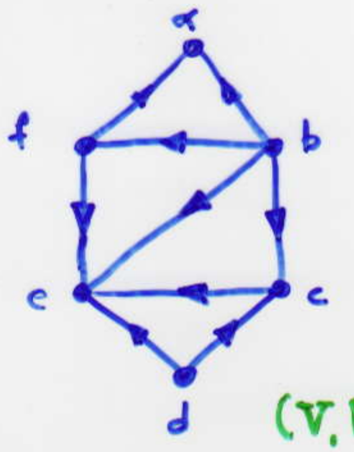
Step I. Label the vertices according to the order determined by $F_1 + F_2$; namely, if $\text{in-degree}(x) = i-1$ then $L(x) = i$.

Step II. Label the vertices according to the order determined by $F_1^{-1} + F_2$; namely, if $\text{in-degree}(x) = i-1$ then $L'(x) = i$

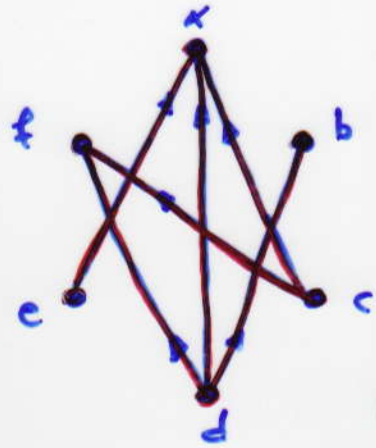
- Notice that

$$xy \in E \Leftrightarrow [L(x)-L(y)]\,[L'(x)-L'(y)] < 0 \qquad (1)$$

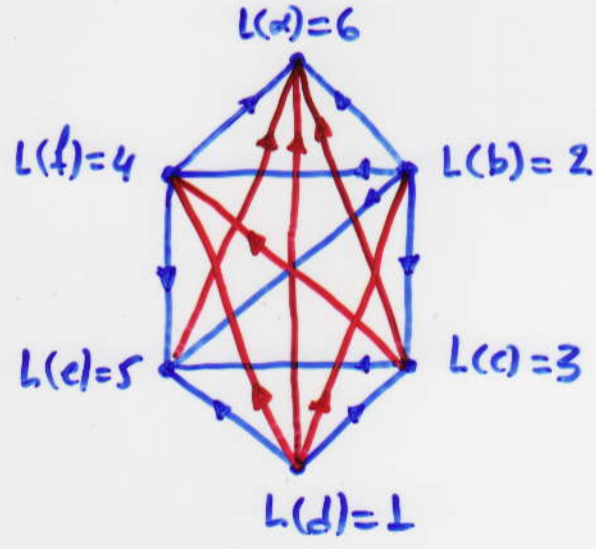
since it is the edges of E which have their orientations reversed between steps I and II.



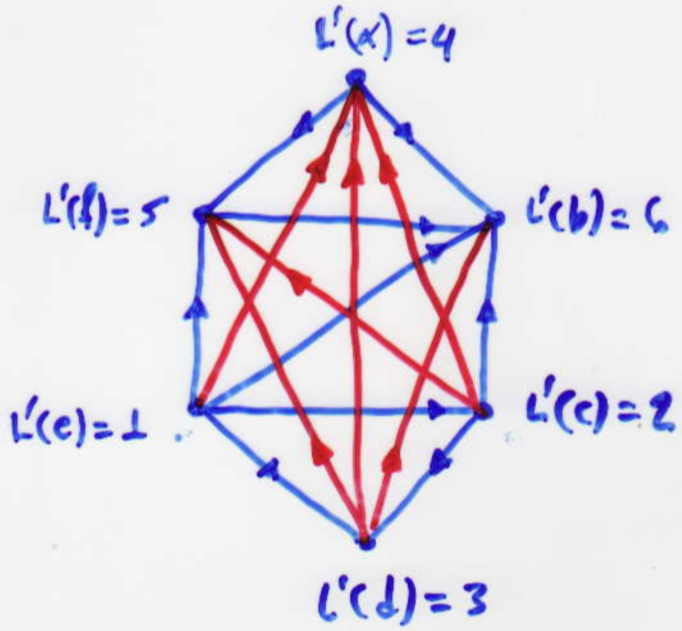
(V, F_1)



(V, F_2)



(V, F_1+F_2)



$(V, F_1^{-1}+F_2)$

- $\pi = [5,3,1,6,4,2]$ (see step III).

Step III. Define π as follows: For each vertex x, if $L(x) = i$, then $\pi_i^{-1} = L'(x)$.

- We have shown that G is a permutation graph if and only if G and $\bar{G}$ are comparability graphs.
- This result suggests an algorithm for recognizing permutation graphs, namely, applying the transitive orientation algorithm to the graph G and to its complement $\bar{G}$.
  - if we succeed in finding transitive orientetions, then the graph is a permutation graph.
- To find a suitable permutation we can follow the construction procedure. The entire method requires $O(n^3)$ time and $O(n^2)$ space for a graph with n vertices.

Permutation Labeling

- A related, but simpler, problem is that of testing whether a given labeling is a permutation labeling.

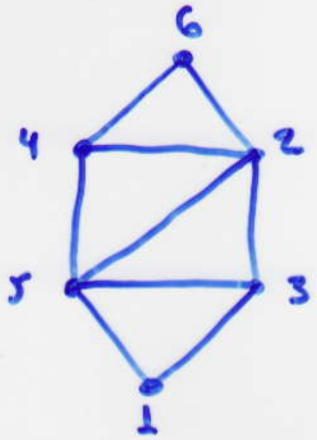
- Let $L : V \longrightarrow \{1, 2, \dots, n\}$ be a bijection labeling of a graph $G = (V, E)$.

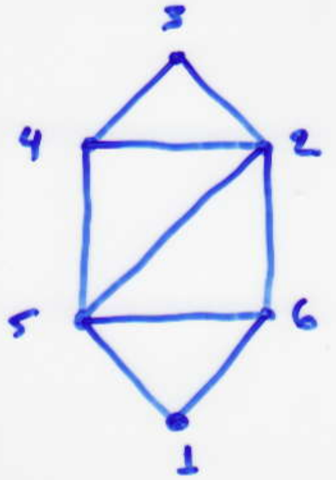
- We call L a permutation labeling if there exists a permutation π of $\{1, 2, \dots, n\}$ such that

$$xy \in E \Leftrightarrow [L(x) - L(y)][\pi^{-1}(L(x)) - \pi^{-1}(L(y))] < 0$$

- Clearly, G is a permutation graph if and only if it has at least one permutation labeling.

- The following figure shows two labelings of the same graph.





This labeling is a permutation labeling

$\{5,3\} - (1) - \{2,4,6\}$

This labeling is not a perm. labeling.

$\{5,6\} - (1) - \{2,3,4\}$

- **Theorem**: Let G be a graph. A bijection $L : V \to \{1,2,\dots,n\}$ is a per. labeling of G iff the mapping

$$F : x \to L(x) - d^-(x) + d^+(x) \quad (x \in V)$$

is an injection, where

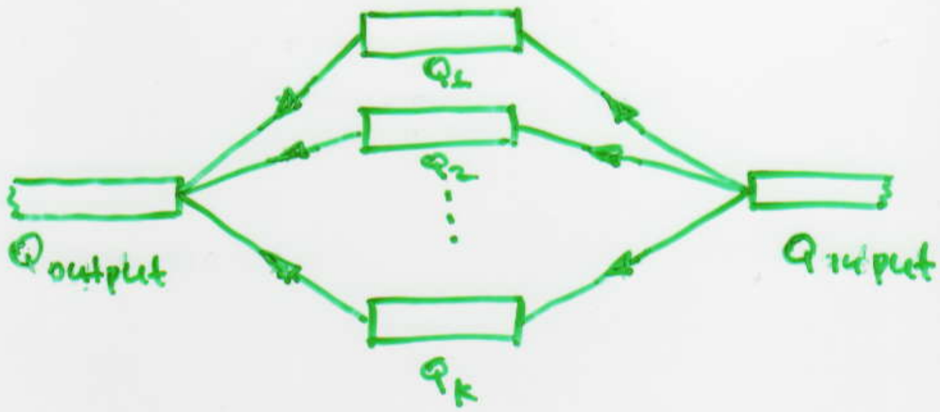
$$d^-(x) = |\{y \in N(x) \mid L(y) < L(x)\}|$$

and

$$d^+(x) = |\{y \in N(x) \mid L(y) > L(x)\}|$$

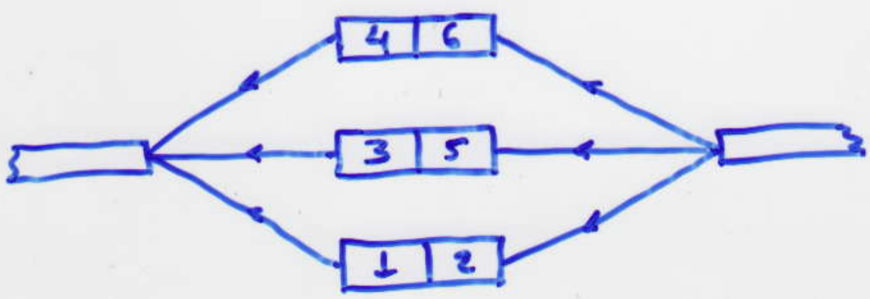
Sorting a Permutation using Queues

- Let us consider the problem of sorting a permution π of $\{1,2,\dots,n\}$ using a network of k queues arranged in parallel.



- Given a permutation π, how many queues will we need?

- Example: $\pi = [4,3,6,1,5,2]$



- What is it that forces two numbers to go into different queues?

  Answer : The numbers occur in reversed order in $\pi = [4,3,6,1,5,2]$

- Thus, if i and j are adjacent in $G[\pi]$, then they must go through different queues.

- **Proposition** : Let $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ be a permutation on N_n. There is a one-to-one correspondence between the proper k-colorings of $G[\pi]$ and the successful sorting stategies for π in a network of k parallel queues.

- **Corollary** : Let π be a permutation on N_n. The following numbers are equal:
  (i) the chromatic number of $G[\pi]$,
  (ii) the minimum number of queues required to sort π,
  (iii) the length of a longest decreasing subsequence of π.

- The canonical sorting strategy for π places each number in the first available queue.

- From this strategy, we obtain the canonical coloring of $G[\pi]$.

- Algorithm Canonical Coloring

Input : A permutation π on N_n;

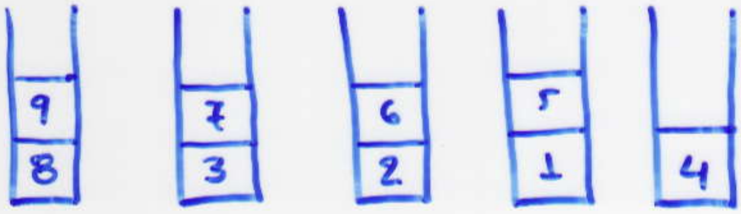
Output : A coloring of the vertices of $G[\pi]$ and $\chi(G[\pi])$;

Method

During the jth iteration, π_j is transferred onto the queue Q_i having the smallest index i satisfying $\pi_j \geq$ last entry of Q_i.

```
begin
    k ← 0;
    for j ← 1 to n do
        i ← first allowable queue;
        COLOR (πj) ← i;
        LAST (i) ← πj ;
        k ← max{k, i};
    end
    χ ← k;
```

- Example: $\pi = [8, 3, 2, 7, 1, 9, 6, 5, 4]$



χ: 1 2 3 4 5

- **Theorem**: Let π be a permutation on N_n. The canonical coloring of $G[\pi]$, as produced by Algorithm Canonical coloring, is a minimum coloring.

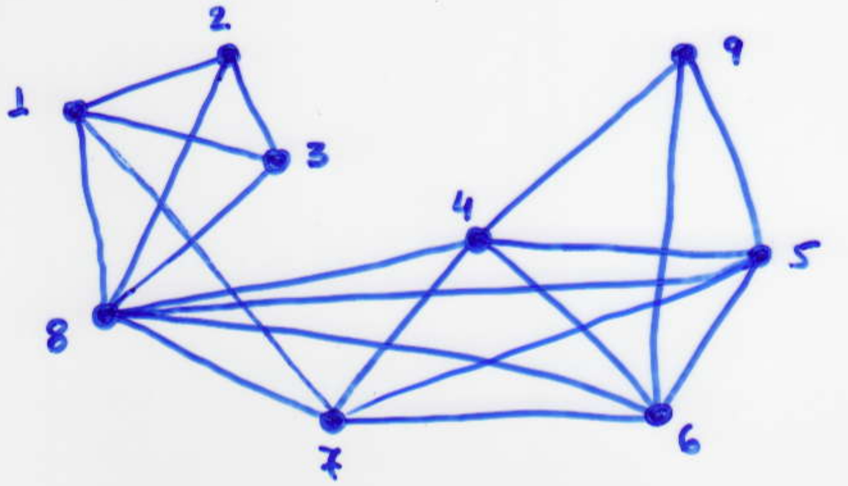
Proof. Clearly the algorithm produces a proper χ-coloring of $G[\pi]$. We must show $\chi = \chi(G[\pi])$. It is sufficient to show that π has a decreasing subsequence of length χ.

Predecessor function p: If $\text{COLOR}(\pi_j) = i \geq 2$, then $\pi_{p(j)}$ = value of $\text{LAST}(i-1)$ during the j-th iteration.

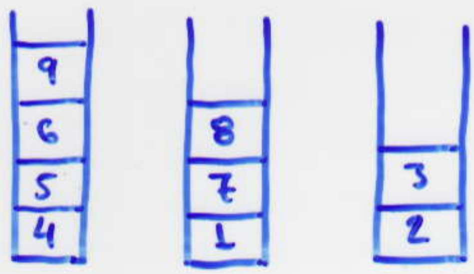
Clearly, $\pi_{p(j)} > \pi_j$ and $p(j) < j$.

Then, $\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_\chi}$ where $\text{COROR}(\pi_{j_\chi}) = \chi$ and $\pi_{j_{i-1}} = \pi_{p(j_i)}$ for $i = \chi, \chi-1, \ldots, 2$.

• **Remark:** To find a minimum clique cover of $G[\pi]$, apply Algorithm canonical coloring to the reversal π^r of π.



$\pi^r = [4, 5, 6, 9, 1, 7, 2, 3, 8]$



• The algorithm canonical coloring runs in $O(n \log n)$ time provided we have the permutation π and the isomorphism $G \to G[\pi]$.

• If we do not have π, then we would revert to the coloring algorithm for comparability graphs.

Basic Properties of Permutations

- Permutations may be represented in many ways.

- The most straightforward is simply a rearrangement of the numbers $1, 2, \ldots, n$, as in the following example:

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
permutation	9	14	4	1	12	2	10	13	5	6	11	3	8	15	7

- An **inversion** is a pair $i < j$ with $\pi_i > \pi_j$. If q_j is the number of $i < j$ with $\pi_i > \pi_j$, then $q_1 q_2 \cdots q_n$ is called **the inversion table of π**.

permutation	9	14	4	1	12	2	10	13	5	6	11	3	8	15	7
inversion table	0	0	2	3	1	4	2	1	5	5	3	9	6	0	8

- The sample permutation given above has **49** inversions.

- A **left-to-right maximum** is an index i:

$$\pi_j < \pi_i \text{ for all } j<i$$

We use the notation $\lambda(\pi)$ to refer to the number of l-r maxima in a permutation π.

- The sample permutation given above: $\lambda(\pi) = 3$.

- A **cycle** is an index sequence $i_1 i_2 \dots i_t$ with $\pi_{i_1} = i_2$, $\pi_{i_2} = i_3, \dots, \pi_{i_t} = i_1$.

We use the notation $(i_1 \, i_2 \dots i_t)$ to specify a cycle.

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
permutation	9	14	4	1	12	2	10	13	5	6	11	3	8	15	7

Cycles $(1\ 9\ 5\ 12\ 3\ 4)\ (2\ 14\ 15\ 7\ 10\ 6)\ (8\ 13)\ (11)$

- An element in π of length n belongs to a unique cycle of length from 1 to n.

- A **derangement** is a permutation with no cycles of length 1.

- If we choose to list the largest element in each cycle (the cycle leaders) first and then take the cycles in increasing order of their leaders, then
  - we get a canonical form that has an interesting property:
    the parentheses are unnecessary !!!
  - since each l-r maximum in the canonical form corresponds to a new cycle.

- Example:

cycles : (1 9 5 <u>12</u> 3 4) (2 14 <u>15</u> 7 10 6) (8 <u>13</u>) (<u>11</u>)

canonical form : 11 12 3 4 1 9 5 13 8 15 7 10 6 2 14

- In combinatorics, this is known as "Foata's correspondence", or the "fundamental correspondence".

- Reference: Sedgewick/Flajolet, "An Introduction to the Analysis of Algorithms", Addison-Wesley, 1996.

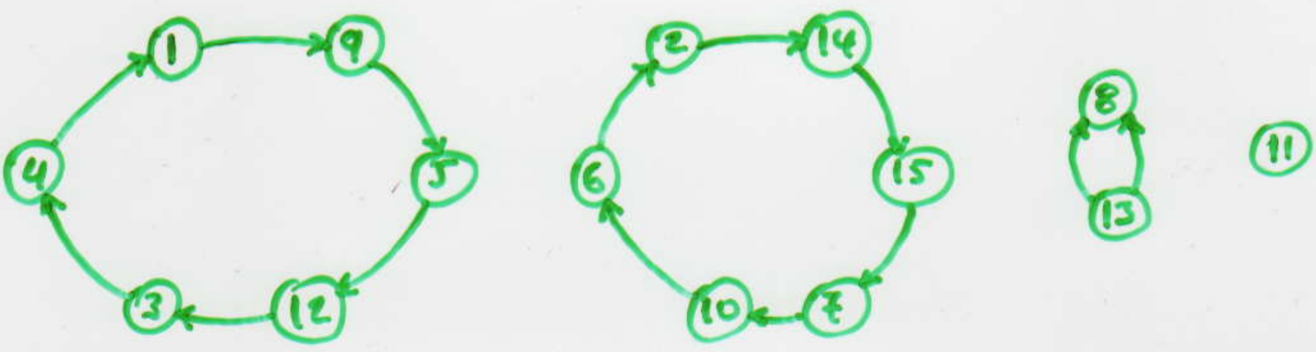
Representations of Permutations

- While it is normally most convenient to represent permutations as rearrangements of the numbers $1, 2, \ldots, n$, many other ways to represent permutations are often appropriate.

Cycle representation

- The fundamental correspondence between a permutation and the canonical form of its cycle structure representation defines a "representation" of permutations.

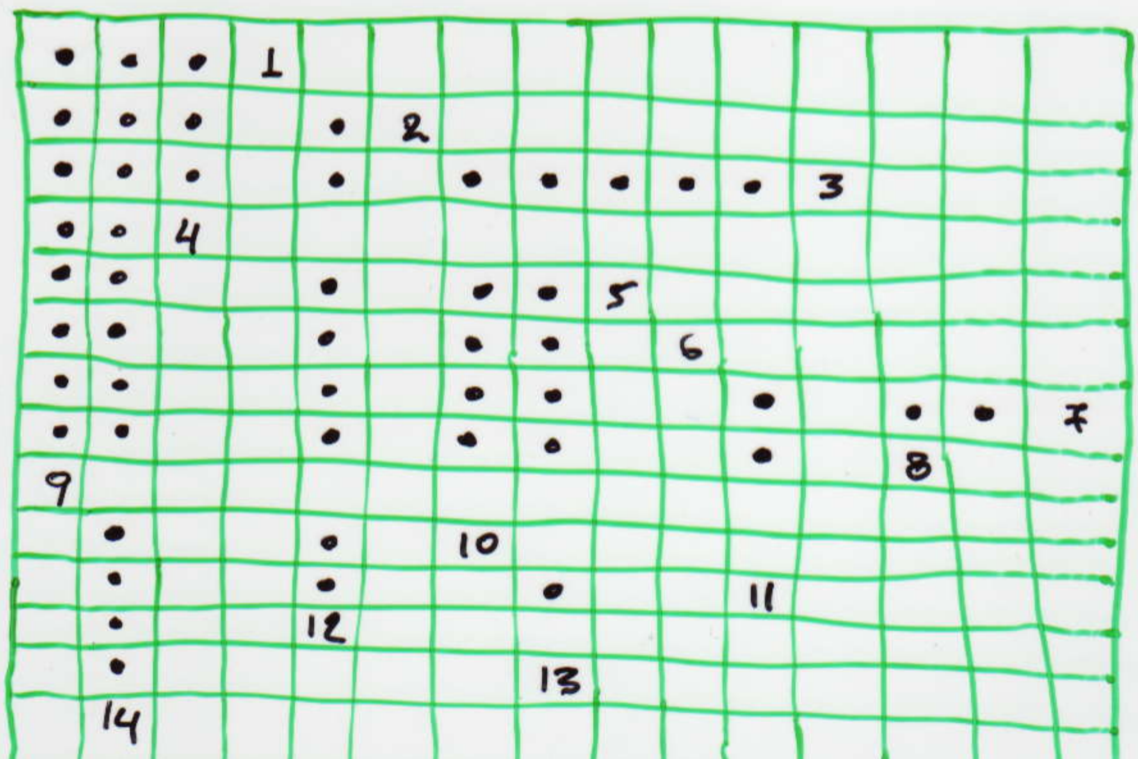
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
permutation :	9	14	4	1	12	2	10	13	5	6	11	3	8	15	7
cycle form :	11	12	3	4	1	9	5	13	8	15	7	10	6	2	14



Lattice representation

- The following figure shows a two-dimentional representation that is useful for studing a number of properties of permutations.

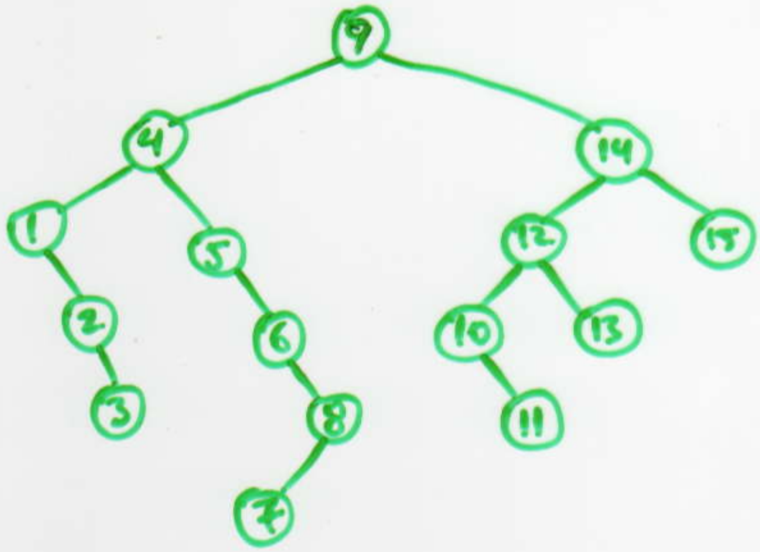
index:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Permutation:	9	14	4	1	12	2	10	13	5	6	11	3	8	15	7



- The permutation π is represented by labelling the cell at row π_i and column i with the number π_i for each i.

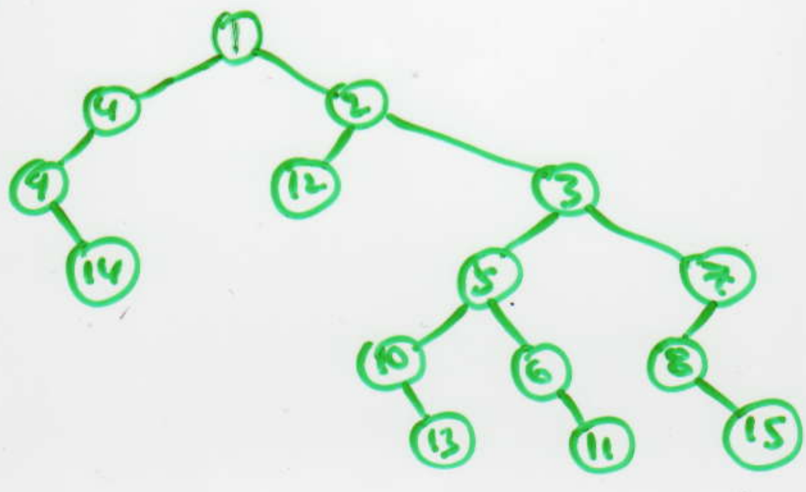
Binary Search Trees:

- A direct relationship between permutations and BST is shown in the following figure:



Heap-ordered Trees:

- A tree can also be built from the lattice repr/tion in a similar maner as illustrated in the next figure:



DAG representation

- We show a strategy to transform a permutation π into a directed acyclic graph (DAG) $G^*[\pi]$ using certain combinatorial properties of π.

- We say that an element π_i inverts π_j if $i<j$ and $\pi_i > \pi_j$.

- An element π_i D-invets π_j if π_i inverts π_j and there exists no element π_k such that π_i inverts π_k and π_k inverts π_j.

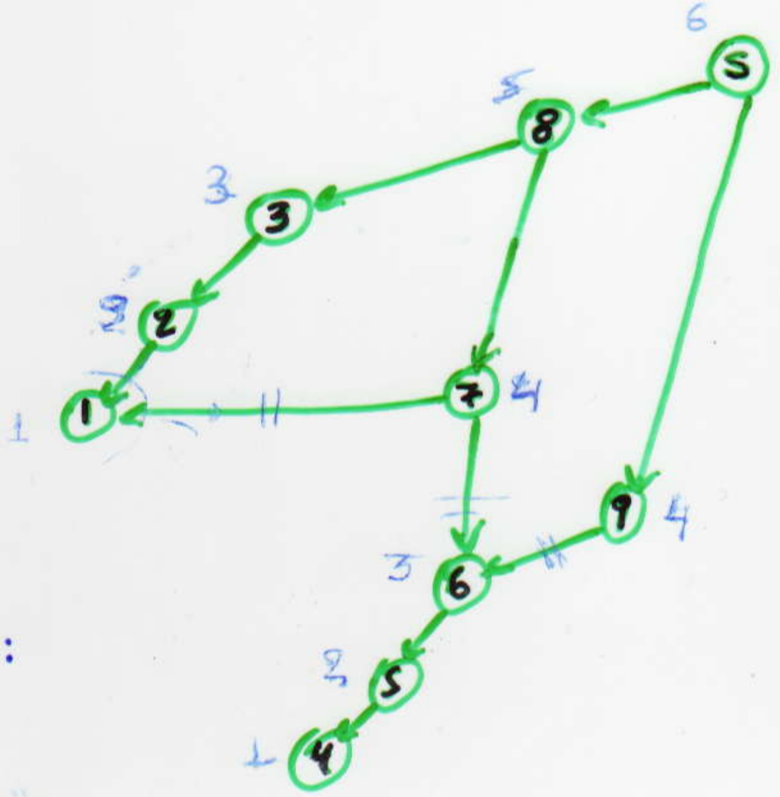
- We construct $G^*[\pi]$ by exploiting the D-inversion relation, as follows:

  1. for every i in π add v_i in $G^*[\pi]$
  2. if i D-inverts j add $\langle v_i, v_j \rangle$ to $G^*[\pi]$
  3. add a dummy vertex s in $G^*[\pi]$ and set $\langle s, v_i \rangle \in E^*$ $\forall$ v_i : indegree$(v_i) = 0$;

- Example

index	1	2	3	4	5	6	7	8	9
Permutation	8	3	2	7	1	9	6	5	4

$G^*[\pi]$:



- We show that there is an 1-1 correspondence between the length of the longest path from s to a vertex v in $G^*[\pi]$ and the color of the vertex v.

- **Lemma**: Let π be a permutation of length n. The following numbers are equal:

  (i) $\chi(G[\pi])$;

  (ii) the length of the longest decr. subseq. of π;

  (iii) the length of the longest path in $G^*[\pi]$;

- Based on these result we can easily show that the problem of coloring a permutation graph $G[\pi]$ can be solved by computing the length $\ell_p(v_i)$ of the longest path $s \rightarrow v_i$ in $G^*[\pi]$.

- This computation can be done in $O(n+m)$ time (how?).

- It is easy to see that the $\chi(G[\pi])$ equals to $\max\{\ell_p(v_i) \mid \text{outdegree}(v_i)=0, 1 \le i \le n\}$.

- Can we compute the lengths $\ell_p(v_i)$ in parallel?

- From $G^*[\pi]$ we can construct a rooted tree $T^*[\pi]$, which we call coloring-permutation tree or cp-tree for short.

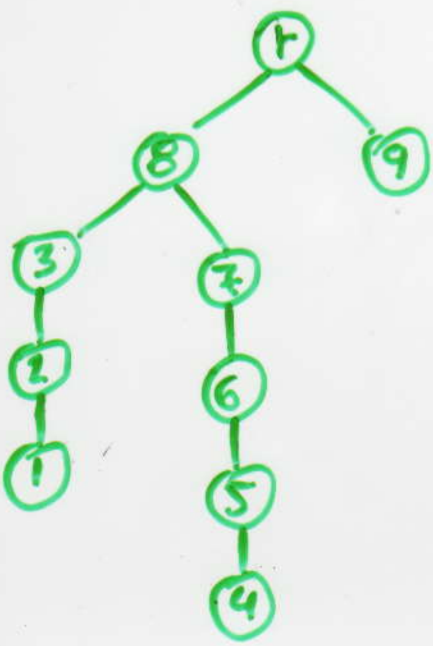
- We define a cp-tree $T^*[\pi] = (V^*, E^*)$ to be a rooted tree having the following properties:

  (i) $V^* = \{r, \pi_1, \pi_2, \ldots, \pi_n\}$, $r > \pi_i \ \forall i$;

  (ii) $\langle \pi_i, \pi_j \rangle \in E^*$ if π_i inverts π_j;

  (iii) there is no pair π_i, π_j : $level(\pi_i) \geq level(\pi_j)$ and π_i inverts π_j;

- The next figure shows the $T^*[\pi]$ of the sample permutation $\pi = [8, 3, 2, 7, 1, 9, 6, 5, 4]$

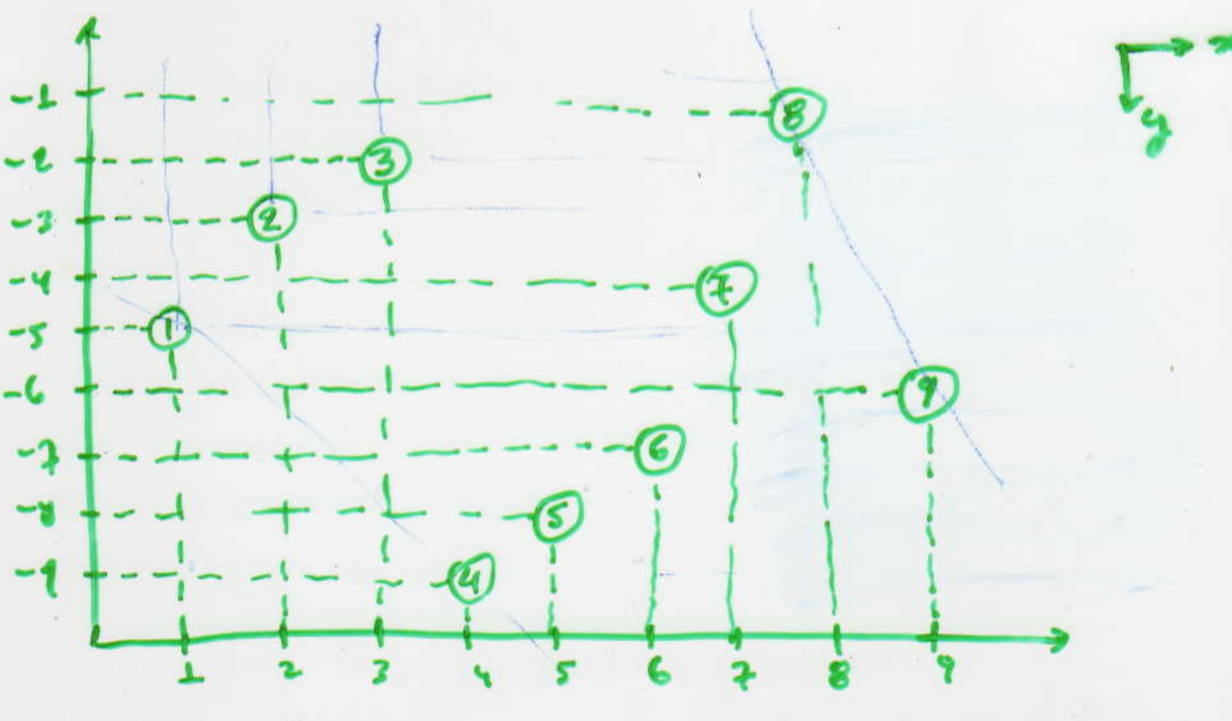


Euclidean plane transformation

- We show a strategy which transforms a permutation π of length n into a set of n points in the plane.

- Let $\pi = [8,3,2,7,1,9,6,5,4]$ be a permutation of length 9. Then,

$$x(i) = i \quad \text{and} \quad y(i) = -\pi^{-1}(i)$$



BFS Tree

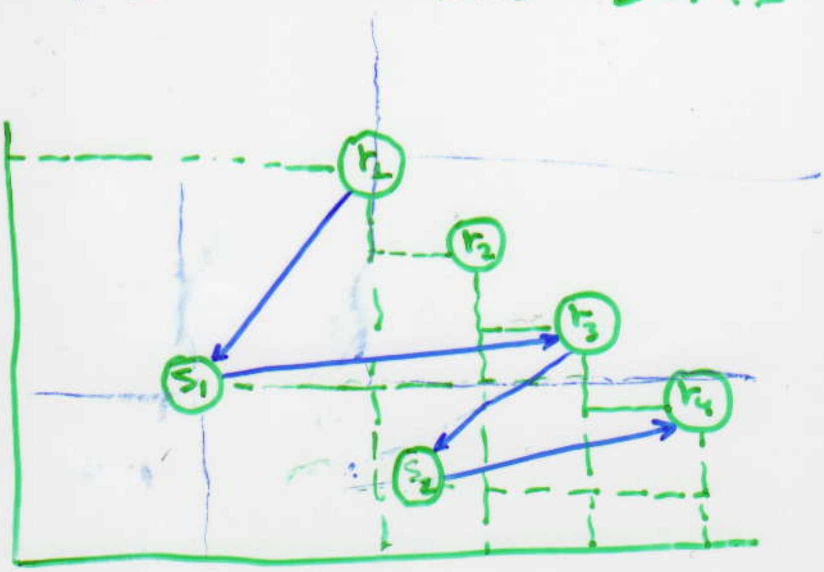
- Let $\pi = [8,3,2,7,1,9,6,5,4]$ be a permutation of lay 9.

- We define two sequences:

  - $R = (r_1, r_2, \dots, r_p)$: the first increasing subseq. going from left to right;

  - $S = (s_1, s_2, \dots, s_q)$: the first decreasing subseq. going from right to left;

- $R[\pi] = [8,9]$ and $S[\pi] = [1,4]$

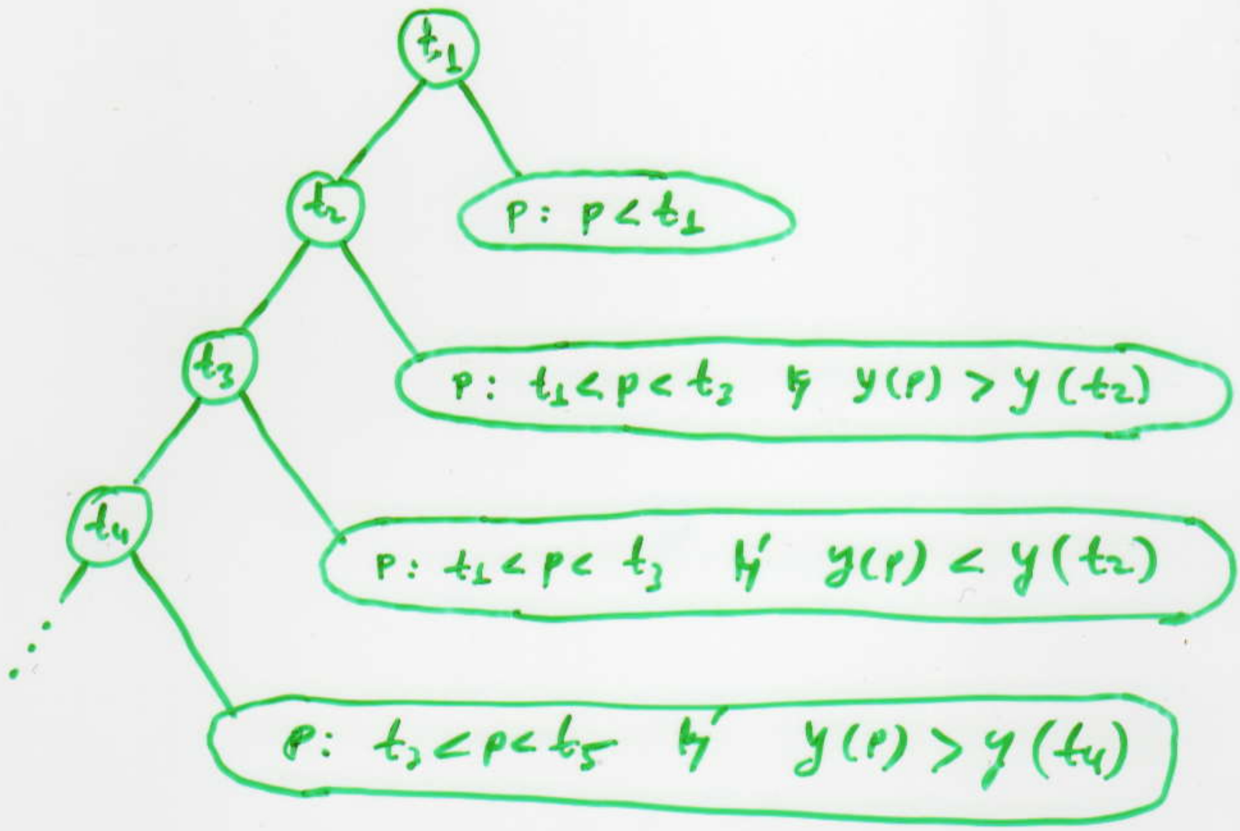
- 

- We constructe the following sequence:

  $r_1, s_1, r_i, \dots, r_p$

- In general: $t_1, t_2, \dots, t_h$

- The BFS tree of the graph $G[\pi]$ has the following structure:



DFS Tree

- Let $G=(V,E)$ be a connected graph and T be a spanning tree of G rooted at $r \in V$.

- T is a depth first spanning tree (DFS) of G if there exist no cross edges.

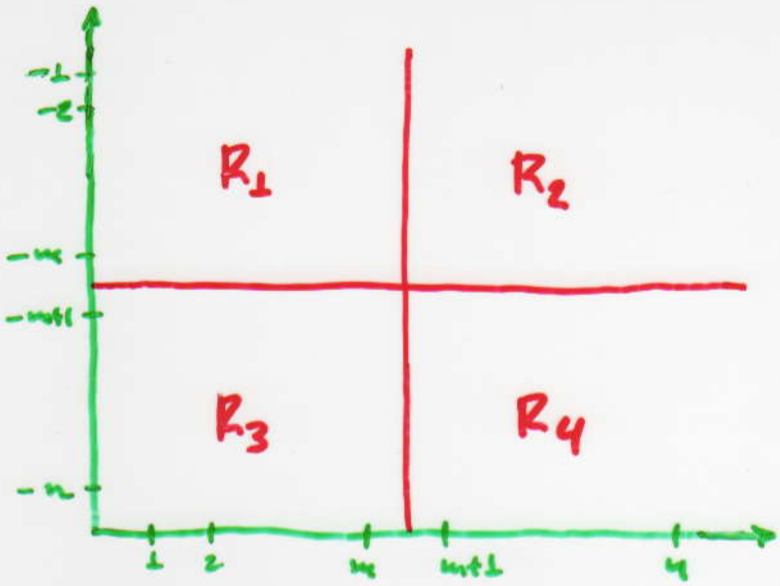
- We show the DFS tree problem can be solved efficiently for permutation graphs.

- Consider the Euclidean plane transformation of a permutation graph G.

- We can solve the problem by computing the decreasing subsequences of π.
  (show how we can do it !!).

- We shall use a different approach.

- Let $m = \lfloor n/2 \rfloor$.



- Draw a horizontal line with y-coordinate

$$-m + 0.5$$

and a vertical line with x-coordinate

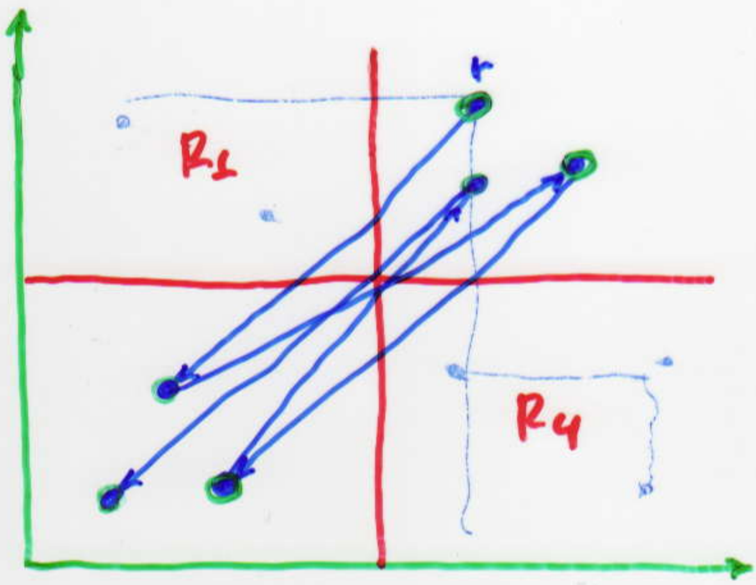
$$m + 0.5$$

- Let n_i $(1 \le i \le 4)$ be the # of vertices in R_i.

- Observe that $n_1 + n_3 = n_1 + n_2 = m$

- This implies $n_2 = n_3$

- Since each vertex in R_2 is adjacent to every vertex in R_3, we can construct a path P consisting of all vertices in R_2 and R_3, where the vertices of P alternate between R_2 and R_3.



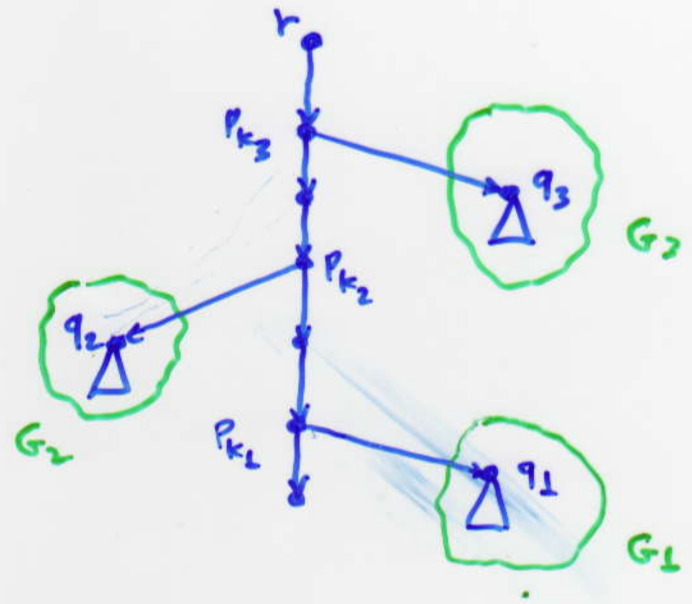
- $P = \{p_1, p_2, \ldots, p_t\}$ starting at $r = p_1$.

- If P is removed from G, the graph $G[R_1 \cup R_4]$ becomes disconnected and each connected component has at most $n/2$ vertices.

- Algorithm DFST

  1. Construct the path $P = \{p_1, p_2, \dots, p_t\}$, where $p_1 = r$ and r the root.

  2. Let $G' = G - P$. Compute the connected components $G_1, G_2, \dots, G_s$ of G'.



  3. For each G_i $(1 \leq i \leq s)$, let p_{k_i} be the vertex of P with the largest index such that p_{k_i} is adjacent to a vertex q_i in G_i.

  4. Recurse on each G_i to find a DFST T_i rooted at q_i.

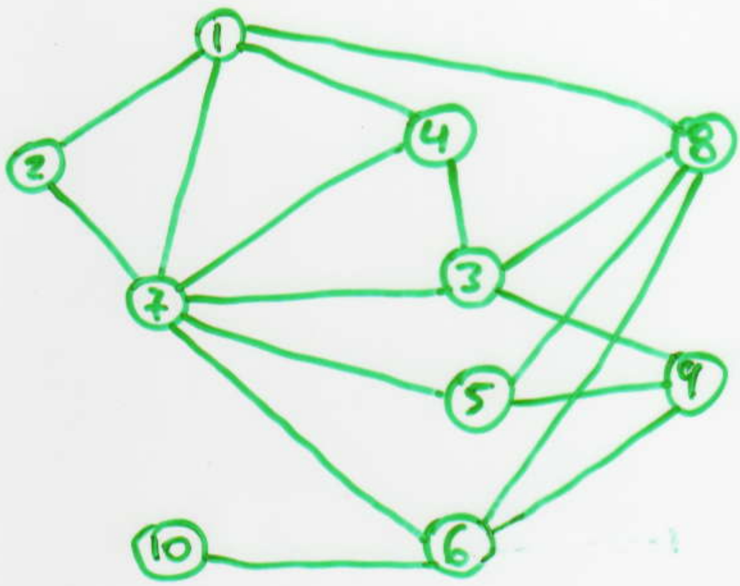
  5. Return $T = P \cup \left(\bigcup_{i=1}^{s} T_i \right) \cup \left(\bigcup_{i=1}^{s} \{(p_{k_i}, q_i)\} \right)$.

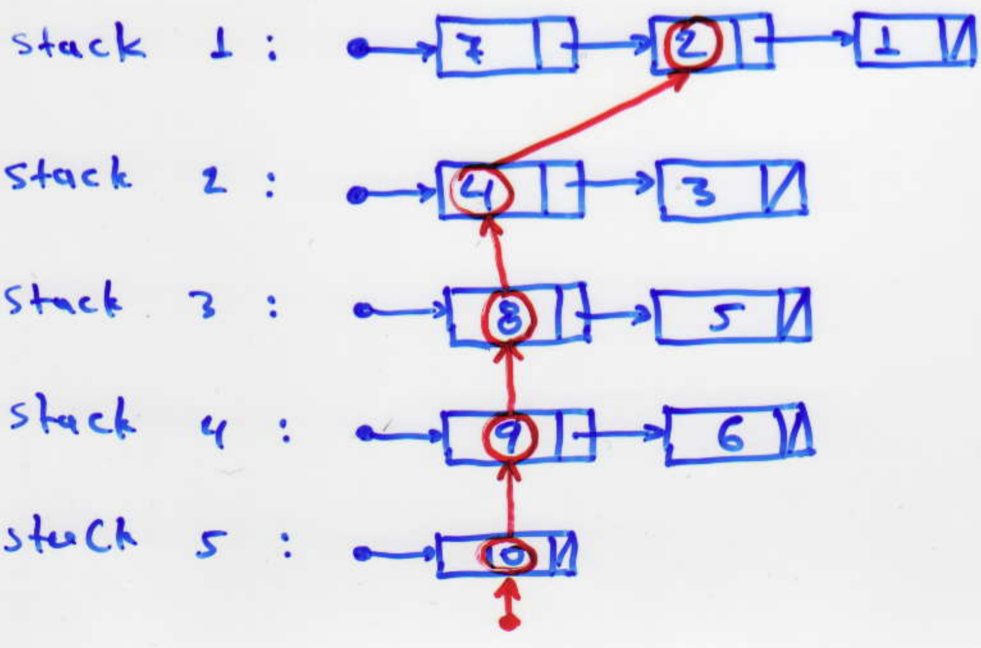
end;

- Hamilton path? Hamilton cycle?

- Maximum indipendent set

$$\pi = [7, 2, 4, 8, 1, 9, 3, 5, 10, 6]$$





- Maximum-weight indipendent set

$\pi = [7, 2, 4, 8, 1, 9, 3, 5, 10, 6]$

$W = [4, 6, 7, 5, 3, 10, 8, 1, 2, 9]$

1. First we have $\pi_1 = 7$, hence the MWIS containing vertex 7 in sequence $[\pi_1] = [7]$ is $I_7 = \{7\}$ and $W_7 = 4$.

2. Secondly we have: $\pi_2 = 2$ in $[\pi_1, \pi_2] = [7, 2]$ is an MWIS containing vertex 2. That is, $I_2 = \{2\}$ and $W_2 = 6$.

continuing in this way, we hav:

$$\left.\begin{array}{ll} I_4 = \{2, 4\} & W_4 = 13 \\ I_8 = \{2, 4, 8\} & W_8 = 18 \\ I_1 = \{1\} & W_1 = 3 \\ I_9 = I_8 \cup \{9\} & W_9 = 28 \\ I_3 = I_2 \cup \{3\} & W_3 = 14 \\ \vdots & \vdots \end{array}\right\} \max W_i \ (1 \le i \le n)$$