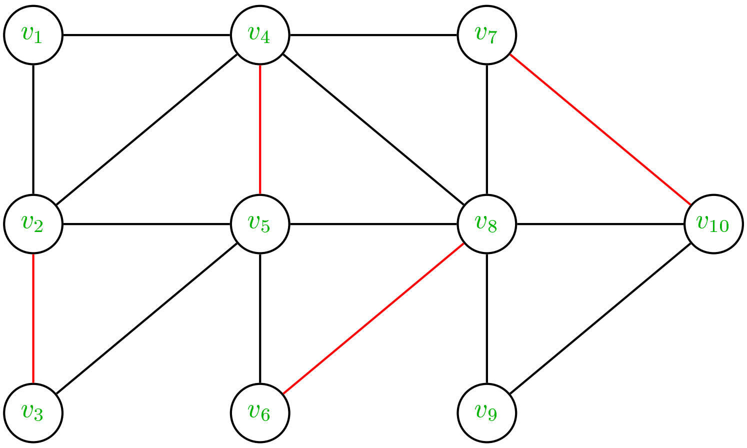
Chapter 2

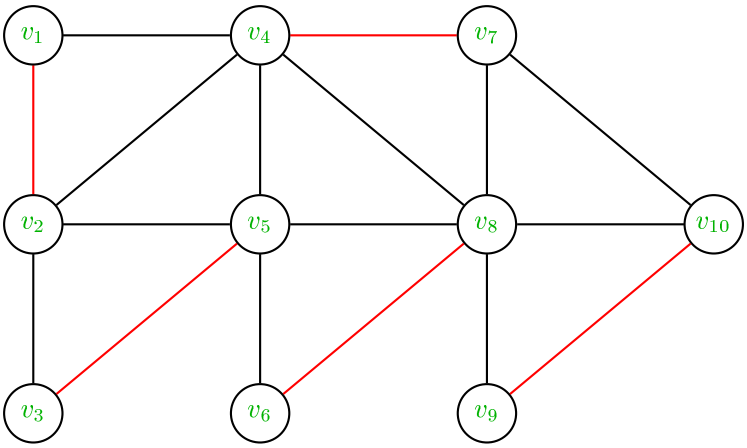
Graph Matching

A matching in a graph $G = (V, E)$ is a set M of edges, $M \subseteq E$, such that no two of the edges of M share a vertex. A vertex $v \in V$ is matched by matching M if some edge in M is incident on v; otherwise, v is unmatched. We are interested in finding a maximum matching M, in other words, a matching of maximum cardinality ($|M| \geq |M'|$ for any other matching M').

Example:



In the above graph, $M_1 = \{(v_2.v_3), (v_4, v_5), (v_6, v_8), (v_7, v_{10})\}$ is a matching.
Is M_1 a maximum matching?



No. $M_2 = \{(v_1, v_2), (v_3, v_5), (v_4, v_7), (v_6, v_8), (v_9, v_{10})\}$ has more edges.

As a matter of fact, M_2 is a maximum matching since any matching of G can never have more than $\frac{|V|}{2} = 5$ edges. □

When the cardinality of a matching M is such that $|M| = \lfloor\frac{|V|}{2}\rfloor$, the largest possible matching in a graph with $|V|$ nodes then matching M is a perfect matching.

Note that in a perfect matching, every vertex $v \in G$ (except maybe one) is matched by M.

The edges of matching M are called matched edges; other edges are free.

If (u, v) is a matched edge, then u is the mate of v.

Vertices that are not incident upon any matched edge are called exposed; the remaining vertices are matched.

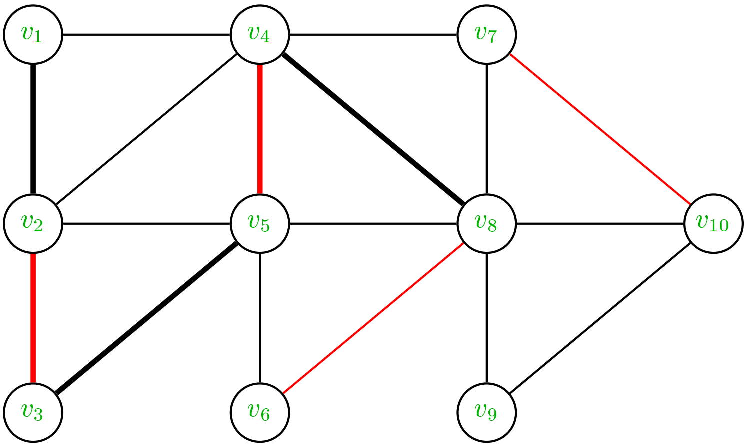
A path $p = \langle u_1, u_2, ..., u_k\rangle$ is called an alternating path if edges $(u_1, u_2), (u_3, u_4)$, ..., (u_{2j-1}, u_{2j}), ... are free, whereas $(u_2, u_3), (u_4, u_5), ..., (u_{2j}, u_{2j+1})$, ... are matched.

Vertices that lie on an alternating path starting with an exposed vertex and have odd rank on this path are called outer. Those with even rank are called inner.

EXAMPLE: Consider the previous graph with matching:

$M_1 = \{(v_2, v_3), (v_4, v_5), (v_6, v_8), (v_4, v_{10})\}$

$P_1 = \langle v_1, v_2, v_3, v_5, v_4, v_8\}$ is an alternating path.



The edges $(v_1, v_2), (v_3, v_5)$ and (v_4, v_8) are free edges, while (v_2, v_3) and (v_4, v_5) are matched edges. P_1 starts with an exposed vertex: v_1.

So v_1, v_3 and v_4 (vertices with odd rank) are called outer, and v_2, v_5, v_8 occupy even positions in P_1 and are therefore inner vertices.

Similarly for $P_2 = \langle v_1, v_4, v_5, v_6, v_8, v_7, v_{10}, v_9\rangle$; it is an alternating path with: v_1 exposed, and outer vertices: v_1, v_5, v_8, v_{10}, and inner vertices: v_4, v_6, v_7, v_9.

$P_3 = \langle v_9, v_8, v_9\rangle$ is also an alternating path; v_9 is exposed, v_9 and v_6 are outer, v_8 is inner. □

An alternating path $P = \langle u_1, u_2, ..., u_k\rangle$ is called augmenting if both vertices u_1 and u_k are exposed.

In our previous example: $P_2 = \langle v_1, v_4, v_5, v_6, v_8, v_7, v_{10}, v_9\rangle$ is an augmenting path.

Recall that if A and B are sets, then $A \triangle B = (A - B) \cup (B - A)$ is the symmetric difference of A and B.

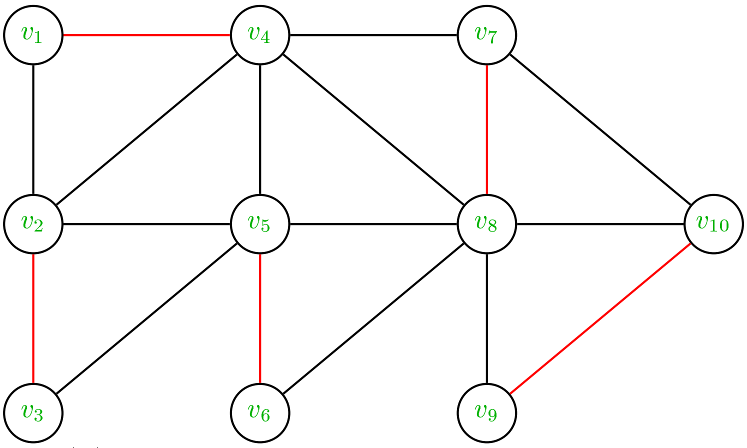
Theorem 2.0.1 *Let P be the set of edges on an augmenting path $p = \langle u_1, u_2, ..., u_{2k}\rangle$ in a graph G w.r.t. the matching M.*
Then $M' = M \triangle P$ is a matching of cardinality $|M|$ + 1.

Note that in our example: $P_2 = \langle v_2, v_4, v_5, v_6, v_7, v_{10}, v_9\rangle$ is an augmenting path w.r.t. M_1.

$P = \{(v_1, v_4), (v_4, v_5), (v_5, v_6), (v_6, v_8), (v_8, v_7), (v_7, v_{10}), (v_{10}, v_9)\}$.
$M_1 = \{(v_2, v_3), (v_4, v_5), (v_6, v_8), (v_7, v_{10}\}$.
$M_1 \triangle P = (M_1 - P) \cup (P - M_1) = \{(v_2, v_3)\} \cup \{(v_1, v_4), (v_5, v_6), (v_8, v_7), (v_{10}, v_{19}\}$.
$\Rightarrow M' = \{(v_2, v_3), (v_1, v_4), (v_5, v_6), (v_8, v_7), (v_{10}, v_{19}\}$.
Note that

1. M' is a matching
2. $|M'| = |M_1| + 1 = 4 + 1 = 5$.

EXAMPLE:



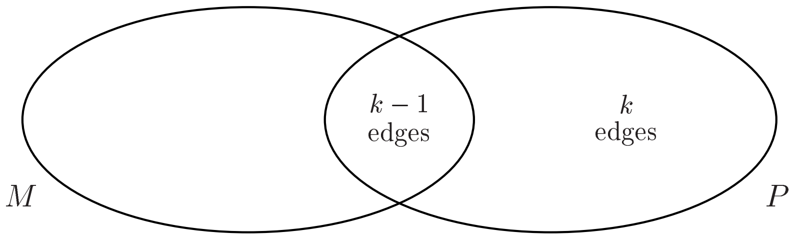
Note that $|M'| = \frac{|V|}{2} = 5$, it does not make any sense to try and look for an augmenting path w.r.t. M'. M' is an maximum matching.
Now let us go back to proving the theorem.

Proof:

1. Show that $M' = M \triangle P$ is a maximum matching.
   We want to show that no two edges in $M \triangle P$ share the same vertex in G.
   The proof is by contradiction.
   Suppose that $e \in M \triangle P$ and $e' \in M \triangle P$ are incident upon the the same vertex.
   Since $M \triangle P = (M - P) \cup (P - M)$, we have 3 cases.
   Case 1: $e \in M - P$ and $e' \in M - P$,
   Case 2: $e \in P - M$ and $e' \in P - M$,
   Case 3: $e \in M - P$ and $e' \in P - M$.
   In case 1, we have two edges, e and e', sharing the same vertex in M. But it is impossible since M is a matching.
   In case 2, since both edges are not in M, they are free edges and both are of the form (u_{2j-1}, u_{2j}) (for different values of j), and therefore cannot be incident to the same vertex.
   In case 3, suppose that $e' \in P - M, e' = (u_{2j-1}, u_{2j})$ is such that u_{2j} is such that u_{2j} is also an endpoint of some edge $e \in M - P$. But u_{2j} is a vertex of the edge $e'' = (u_{2j}, u_{2j+1}) \in M$ (a matching edge).
   So now we have two edges, e and e', in a matching M that share a common vertex. It is impossible.
   So we have just shown that M' is a matching.

2. Show that $|M'| = |M| + 1$ where $M' = M \triangle P$.
   $P = \langle u_1, u_2, ..., u_{2k} \rangle$ is such that u_1 and u_{2k} are exposed;
   $(u_1, u_2), (u_3, u_4), ..., (u_{2j-1}, u_{2j})$ are free (not in M)
   and $(u_2, u_3), (u_4, u_5), ..., (u_{2j}, u_{2j+1})$ are matched (are in M).

Note that there are $2k-1$ edges in P:
$(u_1,u_2),(u_3,u_4),...,(u_{2k-1},u_{2k}),...$ are free, i.e. k of the edges of P are free, and $(u_2,u_3),(u_4,u_5),...,(u_{2k},u_{2k+1}),...$ are in M, i.e. $k-1$ of the edges of P are matched.



$M' = (M-P) \cup (P-M)$ $\qquad$ $|M'| = |M-P| + |P-M|$
(since $(M-P) \cap (P-M) = \emptyset$)

$\Rightarrow |M'| = |M| + 1$ $\qquad$ [$M \cap P$ has one vertex less than $P-M$] $\square$

Note that in the previous theorem we increased the size of matching M by one, to obtain matching M', by inverting the edges of the augmenting path P: The matched edges of M are free edges w.r.t. M', and the free edges of M are now matched edges in M'. (This is true for edges that are on the augmenting path). The edges that are in M and not part of P, will of course be in M'.

Theorem 2.0.2 *A matching M in a graph G is maximum* $\Leftrightarrow$ *there is no augmenting path in G w.r.t. M*

Proof:
$\boxed{\Rightarrow}$ Direct result from the previous theorem. If there is such an augmenting path w.r.t. M then we have $M' = M \triangle P$ a new matching with $|M'| = |M| + 1$ which contradicts the fact that M is a maxmimum matching.
$\boxed{\Leftarrow}$ We have to show:
If there is no augmenting path in G w.r.t. M, then M is a maximum matching. The proof is by contradiction.
Suppose there is no augmenting path in G w.r.t. M and that M is not a maximum matching.
So there exists some matching M' of G such that $|M'| > |M|$.
Consider $M \triangle M' = (M-M') \cup (M'-M)$
The edges in $M \triangle M'$ form a subgraph (that might be disconnected) of G. Since two edges of a matching cannot be incident upon the same vertex, the subgraph $G' = (V, M \triangle M')$ will have the following properties:

1. All vertices will have degree 2 or less. If the degree of a vertex is 2, one of the edges is in M and the other in M'.

2. G' is a collection of cycles and paths.

3. The cycles are of even length and we have the same number of edges in M as in M'.

But since $|M'| > |M|$, it must be the case that in one of the paths, P, we have more edges from M' than from M.
P begins and ends with edges in M', in other words, the first and last vertices of

P are free vertices (w.r.t. M), and so P is an augmenting path in G w.r.t. M. But this is a contradiction since we assumed that G does not have an augmenting path.
Therefore M is a maximum matching. □

Note the big similarity between this theorem and what we have seen with the construction of maximum flows in flow networks.
Edmonds was the first one to show how to find an augmenting path w.r.t. a matching M, in any given graph, in polynomial time.
Edmonds algorithm starts with an empty matching set M, searching for an augmenting path P, increases M by 1 and repeats the process until no augmenting path ca be found.
Next, we restrict our attention to finding maximim matchings in bipartite graphs.

2.1 Maximum Bipartite Matching

A bipartite graph is an undirected graph $G = (V, E)$ in which V is partitioned into two sets $L \subseteq V$ and $R \subset V$ such that

$$(u, v) \in E \Rightarrow \begin{cases} u \in L \text{ and } v \in R \\ \text{or} \\ u \in R \text{ and } v \in L \end{cases}$$

In other words, all the edges of E go between L and R.
The problem of finding a maximum matching in a bipartite graph has many applications.
For instance L: set of matchings,
R: set of tasks to be performed simultaneously.
Edge (u, v) in E: a particular machine $u \in L$ is capable of performing a particular task $v \in R$.
A maximum matching provides work for as many machines as possible.
One way of solving the matching problem for bipartite graphs is to relate it (or reduce it) to the maximum flow problem for flow networks. Thus, any problem that solves the maximum flow problem could now be used to solve the bipartite matching problem.
The flow in the network problem will correspond to matchings in the maximum bipartite matching problem.
Given a bipartite graph $G = (V, E)$ with vertex partition:
$V = L \cup R$, we define the corresponding flow network $G' = (V', E')$ as follows:

a) Add two new vertices, $s \in V'$ and $t \in V'$ and let them be the source and the sink of G'. So $V' = V \cup \{s, t\}$.

b) Add new directed edges from s to the vertices of L, and add new directed edges from the vertices of R to t.

c) Give each edge in the original graph G a direction so that all edges go from L to R.
So $E' = \{(s, u)|u \in L\} \cup \{(u, v)|u \in L, v \in R \text{ and } (u, v) \in E\} \cup \{(v, t)|v \in R\}$.

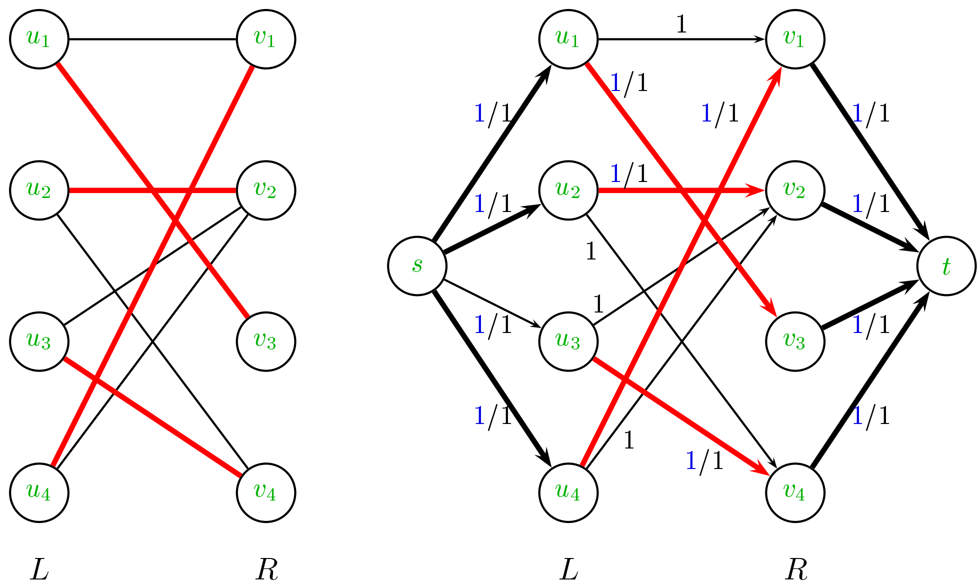
d) Assign unit capacity to each edge in E'.

Example: Given the bipartite graph $G = (V, E)$,
$L = \{u_1, u_2, u_3, u_4\}, R = \{v_1, v_2, v_3, v_4\}$ with edges shown in the figure.

The figure also shows a maximum matching, the corresponding flow network G'.

$$M = \{(u_1,v_3),(u_2,v_2),(u_3,v_4),(u_4,v_1)\} \qquad\qquad |f| = \sum_{i=1}^{4} f(s,u_i) = 4.$$

$$|M| = 4$$



If $f(u,v)$ is an integer for all $(u,v) \in V \times V$ then we say that the flow f is integer-valued.

Theorem 2.1.1 *$G = (V,E)$ is a bipartite graph with vertex partition $V = L \cup R$. $G' = (V',E')$ is the corresponding flow network.*

a) M is a matching in G $\Rightarrow$ there is an integer-valued flow f in G' with $|f| = |M|$,

b) f is an integer-valued flow in G' $\Rightarrow$ there is a matching M in G with $|M| = |f|$.

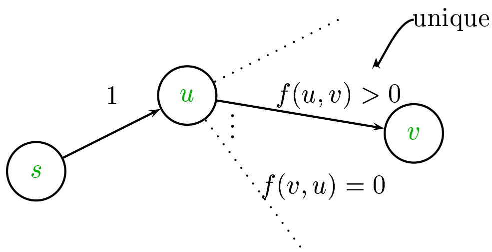
Proof:

a) Suppose that M is a matching in G.
Define the following flow function f in the corresponding flow network G':

$$(u,v) \in M \quad \Rightarrow \quad \begin{array}{l} f(s,u) = f(u,v) = f(v,t) = 1 \\ f(u,s) = f(v,u) = f(t,v) = -1 \end{array}$$

In other words, each edge in $M, (u,v) \in M$, corresponds to 1 unit of flow in G' that goes along the path $s \to u \to v \to t$.
For all other edges $(u,v) \in E'$, we define $f(u,v) = 0$.
It is easy to show that f thus defined satisfies the three properties of:
skew symmetry, capacity constraint and flow conservation.
Note that we have a cut $(L \cup \{s\}, R \cup \{t\})$, in the flow network G' and that the capacity of the cut is such that:
$C(L \cup \{s\}, R \cup \{t\}) = |M|$.
By a previous theorem we know that $c(S,T) = |f|$ for any cut (S,T).
So we have that the value of the flow we defined: $|f| = |M|$.

b) Let f be an integer-valued flow in G'. We need to define a matching M such that: $|M| = |f|$.
Define $M = \{(u,v)|u \in L, v \in R \text{ and } f(u,v) > 0)\}$.
First, we show that M is a maximum matching in G.



From the construction of G' we have that: Each $u \in L$ has one entering edge (s,u) and its capacity is 1.
Vertex u has at most one unit of positive net flow entering it.
Flow function f is integer-valued $\Rightarrow$ for each $u \in L$:

one unit of positive net flow enters u	$\Leftrightarrow$	there exists one $v \in R$ such that $f(u,v) = 1$

Thus, at most one edge leaving each $u \in L$ carries positive net flow.
Similarly for each vertex v in R; there is at most one edge entering each vertex $v \in R$ with positive net flow.
So $M = \{(u,v)|u \in L, v \in R \text{ and } f(u,v) > 0\}$ is indeed a matching in G.
Second, we show that $|M| = |f|$.
But $R = V' - (\{s,t\} \cup L)$, and by using properties studied in the section on maximum flow:
$f(L,R) = f(L,V') - f(L,L) - f(L,s) - f(L,t)$ and
$f(L,V') = 0, f(L,L) = 0, f(L,t) = 0$ (no edges between L and t)
$\Rightarrow |M| = -f(L,s) = f(s,L) = f(s,V') = |f|$ □

So to compute a maximum matching in G, we run the Ford-Fulkerson algorithm in G'. The integrality theorem:

> *If all capacities are integers, then the maximum flow f produced by the Ford-Fulkerson algorithm is such that $|f|$ is integer-valued and all $f(u,v)$ have integer values;*

assures the success of finding a matching in G.
Moreover, it is not difficult to show that the cardinality of a maximum matching in a bipartite graph G is the value of a maximum flow in its corresponding flow network G'.
As for the running time of finding a maximum matching in a bipartite graph, the complexity is $O(|V||E|)$.
Any matching in a bipartite graph has cardinality at most $min(L,R) = O(|V|)$, and since $|f| = |M|$, the value of the maximum flow in G' is $O(|V|)$.

Bipartite problems are easy to solve since they can be modeled as network flow problems and thus can be solved by using any of the many algorithms we have studied.
Note that the resulting network flow problem has a special structure, and thus one can refine algorithms with better worst-case complexity for bipartite matching problems than normal network problems.
Nonbipartite matching problems are more difficult to solve because they cannot be transformed to standard network flow problems. Nonbipartite matching problems require specialized combinatorial algorithms.

In the next section we cover some applications that can be found in [AMB93] [Wes96] and [CLR90].

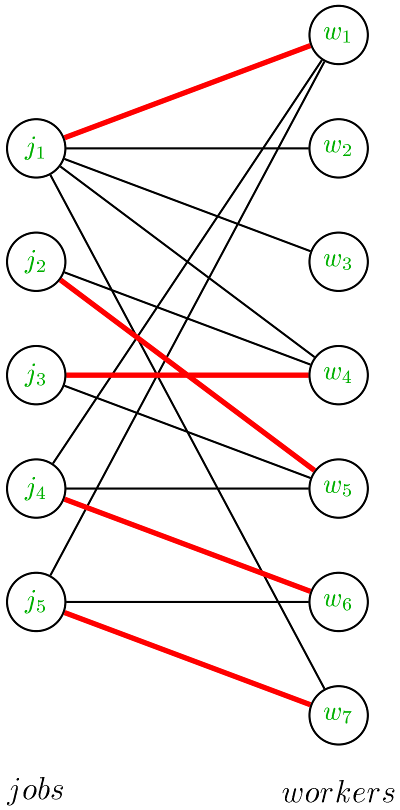
2.2 Applications

2.2.1 Bipartite Personnel Assignment

In many problems we wish to assign people to objects: e,g, jobs, machines, rooms or even each other.

a) The Personnel-assignment problem:
A company requires a number of different jobs, and each worker is suited for some of these jobs, but not others.
Assuming that each person can perform at most one job at a time, how should the jobs be assigned so that the maximum number of jobs can be performed simultaneously?

EXAMPLE:



The maximum matching whose edges appear in red shows that all five jobs can be assigned to workers.
There are 5 jobs and 7 workers. Each edge represents a suitable job assignment. How can we assign the jobs so that the maximum number of jobs can be performed simultaneously?

b) A swimming coach must select from her best 8 swimmers a medley relay team of four, each of whom will then swim one of the 4 stroaks: back, breast, butterfly and free-style. The coach knows the time of each swimmer in each stroke. The problem is to identify the team of the 4 best swimmers out of the eight that are available. The sum of times obtained by optimally matching four strokes gives the minimum feasible relay time and the corresponding team is the best team.

c) Assigning medical school graduates to hospitals
Thousands of doctors graduate from medical school in the USA. They are then eligible for residences at the different hospitals around the country. The American medical association (AMA) performs a matching process in which the graduates rank the hospitals according to their preferences and the hospitals rank the graduates according to their preferences. The AMA assigns the graduates to hospitals so that the matching is "stable".
An assignment is unstable if some graduate i is nor assigned a hospital j. But the graduate prefers hospital j over his/her current assignment, and at the same time, hospital j prefers graduate i over one of the graduates assigned to it. This alignment is unstable because both the graduate i and the hospital j have an incentive to change their current assignments. An assignment that is not unstable is said to be stable.
The objective of the AMA is to identify a stable assignment.

2.2.2 Non bipartite Personnel Assignment

Pairing volunteers for a rescue mission
Several first-aid workers from around the world have volunteered for a rescue mission in some third-world country that has been struck by a disaster. The volunteers are to be divided into two-person teams, but the members of each team must speak the same language. What is the maximum number of pairs that can be sent out on the rescue mission?
Use a graph where each vertex represents a volunteer, and an edge between volunteers i and j means that both speak the same language. A maximum matching in the graph corresponds to a maximum set of two-person teams.

□

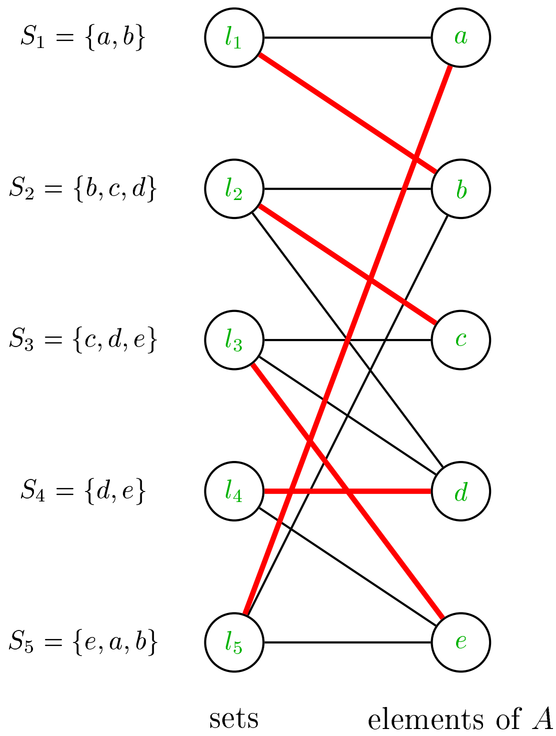
2.3 Transversals for families of subsets

Suppose that A is a nonempty finite set, and $F = \{S_1, S_2, ..., S_r\}$ is a family of (not necessary distinct) nonempty subsets of A.
A transversal (or system of distinct representatives) of family F is a sequence $T => a_1, a_2, ..., a_r >$ of r distinct elements of A, such that:
$a_i \in S_i$ for each $i = 1, 2, ..., r$.

Example: $A = \{a, b, c, d, e\}$
$F = \{S_1, S_2, S_3, S_4, S_5\}$ where:
$S_1 = \{a, b\}, S_2 = \{b, c, d\}, S_3 = \{c, d, e\}, S_4 = \{d, e\}$ and $S_5 = \{e, a, b\}$. Then $T =< b, c, d, e, d, a >$ is a transversal, or a system of distinct representatives for family F. □
Note that to solve the "assigning medical school graduates to hospitals" application is in essence reduced to finding a transversal for the family $F = \{S_1, S_2, ..., S_r\}$ where:
S_i: is the set of all hospitals that find applicant i acceptable, and we have r graduates who applied for residences.
In this application, we assume that we do not have any ranking of hospitals by the graduates or any ranking of graduates by hospitals. If we do take the ranking into consideration, then one has to check the transversal to see if the assignment is stable.
Suppose that $G = (V, E)$ is a bipartite graph with vertex partition:
$V = L \cup R$, and that M is a matching in G.
Matching M is said to be L-saturating if each vertex in L is an endpoint of an edge

in M.
Similarly, M is R-saturating if each vertex in R is an endpoint in M.
Note that every L-saturating matching and every R-saturating matching must be a maximum matching, since in either case, either all vertices of L are matched (L-saturating) or all vertices of R are matched (R-saturating).
An L-saturating matching is also known as a complete matching from L into R.
Given a set $A = \{a_1, a_2, ..., a_r\}$ and a family $F = \{S_1, S_2, ..., S_r\}$ of (not necessarily distinct) subsets of A, we construct a bipartite $G = (V, E)$ with vertex partition $V = L \cup R$ in the following fashion:
$L = \{l_1, l_2, ..., l_r\}$ where each l_i corresponds to subset S_i in F, and $R = \{a_1, a_2, ..., a_n\}$ (i.e. $R = A$), and an edge exists between l_i and a_j if $a_j \in S_i$.
Then a transversal for family F corresponds to an X-saturating matching of graph G.

Example: Consider our previous example where $A = \{a, b, c, d, e\}, F = \{S_1, S_2, S_3, S_4, S_5\}$ and $S_1 = \{a, b\}, S_2 = \{b, c, d\}, S_3 = \{c, d, e\}, S_4 = \{d, e\}$ and $S_5 = \{e, a, b\}$.



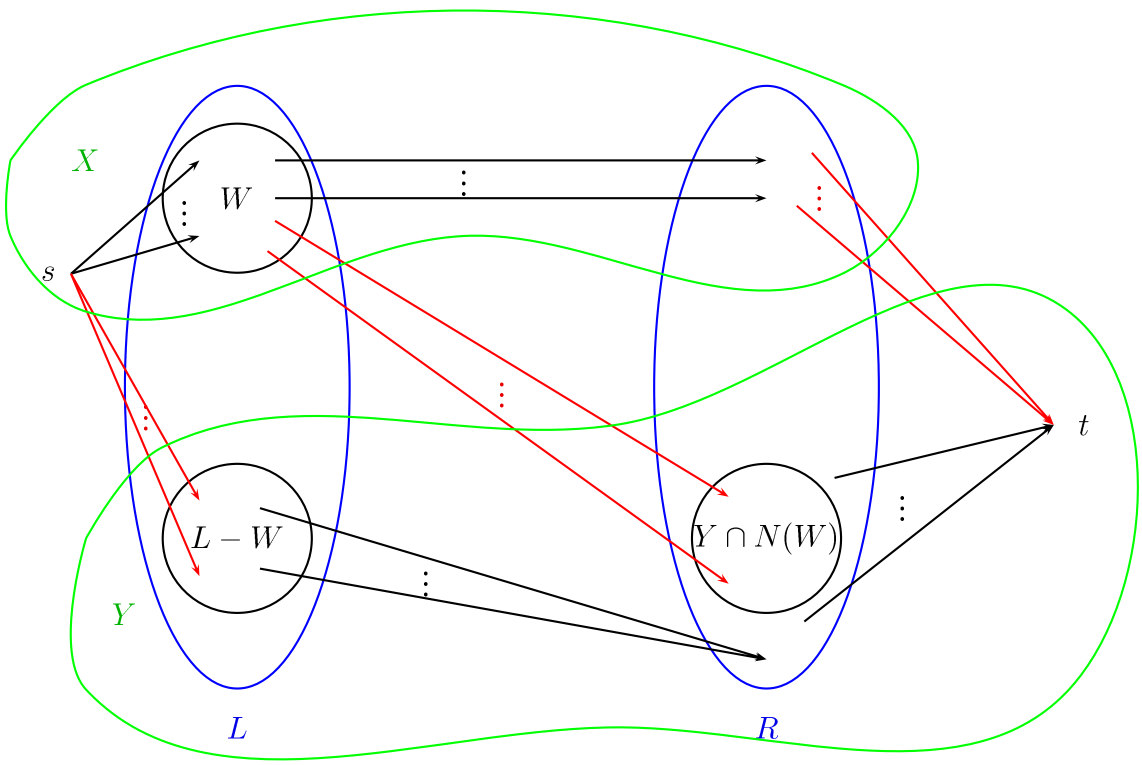
L-saturating matching : $\{(l_1, b), (l_2, c), (l_3, e), (l_4, d), (l_5, a)\}$ that corresponds to the transversal $T =< b, c, e, d, a >$ □

2.4 Hall's theorem

Hall's theorem gives a necessary and sufficient condition for a bipartite graph to have an L-saturating matching.
The theorem makes use of the set of neighbors of a subset of vertices.
Suppose that $G = (V, E)$ is a bipartite graph with vertex partition $V = L \cup R$.
Let $W \in L$. Then $N(W)$ denotes the subset of R consisting of all vertices in R that are adjacent to at least one vertex in W.
$N(W)$ is the set of neighbors of W.
The following is the 1935 theorem for bipartite graphs by Hall.

Theorem 2.4.1 *Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$.*
G has an L-saturating matching $\Leftrightarrow$ for each subset $W of L$, $|W| \leq |N(W)|$.

Proof:
$\boxed{\Rightarrow}$ Suppose that we have an L-saturating matching M in G.
Let W be any subset of L. Since M is L-saturating, every vertex $w \in W$ is matched to a vertex $y \in R$ by an edge in M.
Each such y is in $N(w)$.
So $|W| \leq |N(W)|$.
$\boxed{\Leftarrow}$ Suppose that Hall's condition holds; i.e. $|W| \leq |N(W)|$ for every $W \subseteq L$.
Construct the corresponding flow network: $G' = (V', E')$.
Note that an L-saturating matching in G would be a maximum matching in G and would correspond to a maximum flow in G' with value $|L|$.
By the max-flow min-cut theorem, it suffices to show that a minimum cut in G' has capacity $|L|$.
We know, by the construction of G' itself, that:
$cut(\{s\}, V' - \{s\} - \{t\})$ has capacity $|L|$, since all edges leaving s to L have capacity one.
So the only thing we have to prove is that any other cut of $G', (X, Y)$, will be such that:
capacity $(X, Y) \geq |L|$.
Let (X, Y) be any cut in G', and let $W = X \cap L$.
Then $cut(X, Y)$ can be expressed as the disjoint union of three arc sets.



Note that all the edges of $cut(X, Y)$ are in 3 sets:
C_1: the set of edges from s to $L - W$.
C_2: the set of edges from W to $Y \cup N(w)$.

C_3: the set of edges from $X \cap R$ to t.
Note that these edges are red in the figure.
So $cut(X,Y) = C_1 \cup C_2 \cup C_3$.
$\Rightarrow$ capacity of $cut(X,Y) = |C_1 + |C_2| + |C_3|$ (since the sets are pairwise disjoint)
$= |L - W| + |C_2| * |X \cap R|$ (by construction of $G'-$ all capacities $= 1$).
$|C_2| \geq |Y \cap N(w)|$ (by definition of neighbors)
$= |N(w)| - |X \cap N(w)|$
Since $N(w) = [N(W)| - |X \cap N(w)] \cup [Y \cap N(w)]$.
So $|C_2| \geq |N(w)| - |X \cap N(w)|$ and:
capacity of $cut(X,Y)$
$\geq |L - W| + |N(W)| - |X \cap N(w)|X \cap R|$
Since $N(w) \subseteq R$, we have: $|X \cap N(w)| \leq |X \cap R|$.
Therefore: capacity of $cut(X,Y)$
$\geq |L - W| + |N(W)|X \cap R| + |X \cap R|$
$= |L - W| + |N(W)|$
But we assume that Hall's condition holds.
So $|N(W)| \geq |W|$.
Capacity of $cut(X,Y) \geq |L - W| + |N(W)| \geq |L - W| + |W| = |L|$
$\Rightarrow$ Minimum cut in G' has capacity $= |L|$ $\square$

Hall's theorem can be restated in terms of a family of subsets, giving the necessary and sufficient condition for the family to have a transversal.
Obviously, the proof is exactly the same as the one we just saw.
Hall's theorem for transversals:
Let A be a nonempty finite set, and $F = \{S_1, S_2, ..., S_r$ be a family of nonempty subsets of A.

family F has a transversal $\Leftrightarrow$ the union of any k of the subsets S_i contains at least k elements of A, for $1 \leq k \leq r$.

One of the first applications of Hall's theorem is the marriage problem:
Given a set of women, each of whom knows a subset of men, under what conditions can each of the women marry a man whom she knows?
Note that this problem has many variations.
For example, if $|L| = |R|$, Hall's theorem is known as the "marriage theorem".
The name arises from the scenario of a symmetric compatibility relation between a set of n men and a set of n women. If every man is compatible with k women and every women is compatible with k men, then there must be a complete matching using compatible pairs.

We know that a complete matching is a maximum matching.
But what if G does not have a complete matching? How can we show that a certain matching is maximum? We know that:

A matching M in a graph G is a maximum matching in G $\Leftrightarrow$ G has no augmenting path w.r.t. M

But this is not very practical: exploring all alternating paths w.r.t. M would take a very long time.
We wold rather find an explicit structure in G that forbids a matching larger than M.

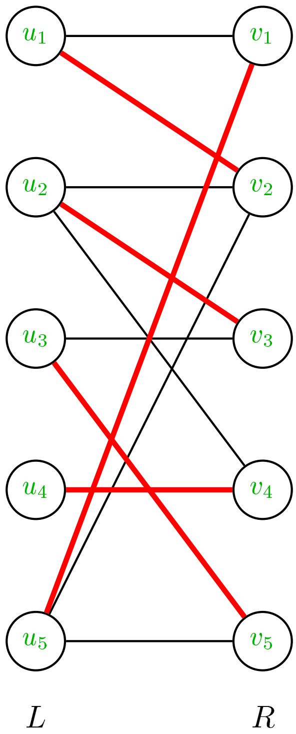
2.5 Maximum Matchings and Minimum Vertex Covers

A vertex cover of a graph $F = (V, E)$ is a set S of vertices, $S \subseteq V$, such that S contains at least one endpoint of every edge of G. We say that the vertices in S cover the edges of G.
A minimum vertex cover is a vertex cover with the least number of vertices.

EXAMPLES:

1. If the graph G represents a road network in some village that contains no traffic lights, and such that the roads are straight (and contains no isolated vertices), then the problem of assigning the minimum number of police person in order to observe and direct traffic in the entire road network is in essence solvable by finding the minimum vertex cover of the network.

2. Consider the following bipartite graph $G = (V, E)$ with vertex partition $V = L \cup R$.

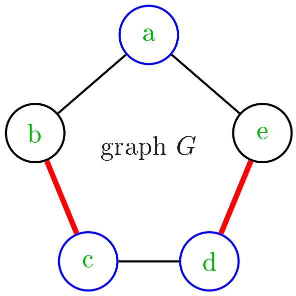


Matching $M = \{(u_1, v_2), (u_2, v_3), (u_3, v_5), (u_4, v_4, (u_5, v_1)\}$ is a maximum matching. $|M| = 5$.
$S = \{v_1, v_2, v_3, v_4, v_5\}$ is a minimum vertex cover of G. $|S| = 5$.

IF M is a matching in a graph G and S is a vertex cover of G, then $|M| \leq |S|$.
This is due to the fact that no two edges e and e' of M can be covered by a single vertex v, and so the size of S is at least the size of M.
Consequently, if one can find a matching M in a graph G and a vertex cover S such that $|M| = |S|$, then we know that M is a maximum matching and that S is a minimum vertex cover.
We can always find a maximum matching M and a minimum vertex cover S (with

$|M| = |S|$) for bipartite graphs.
But that does not hold for nonbipartite graphs.

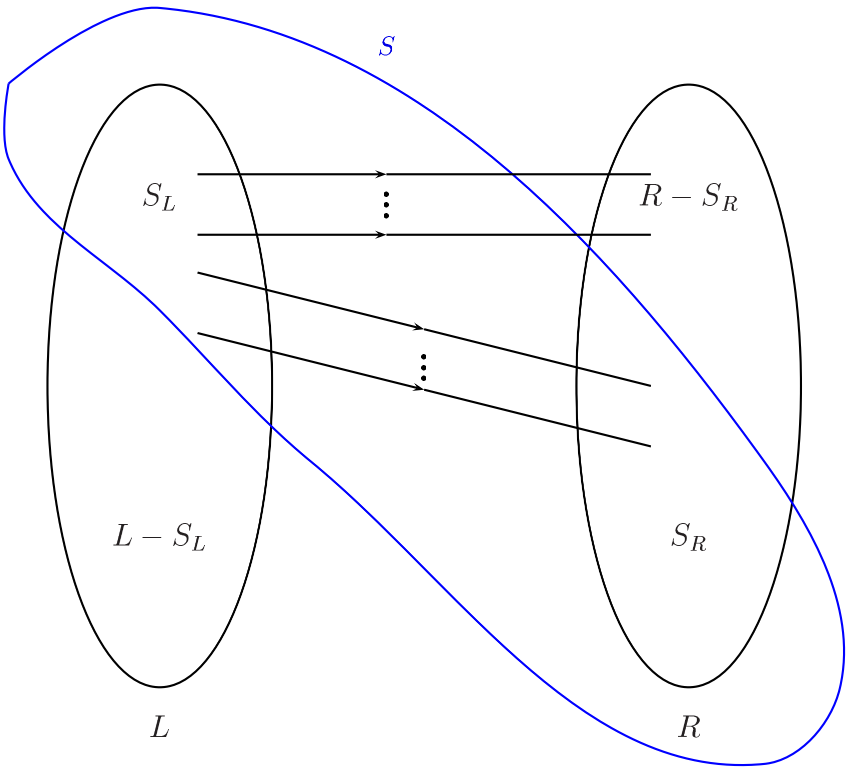
EXAMPLE: In graph G: $M = \{(b, c), (d, e)\}$ is a maximum matching.
$S = \{a, c, d\}$ is a minimum vertex cover and $|M| < |S|$.



Next we formally prove that for bipartite graphs, the size of maximum matchings and minimum vertex covers are identical.
The theorem is due to König.
König's theorem is a min-max theorem since it states equality between answers to a minimization problem and a maximization problem.

Theorem 2.5.1 *Theorem (König)*
Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$. The number of edges in a maximum matching in G is equal to the number of vertices in minimum vertex cover of G.

Proof:
Let S be a minimum vertex cover of G.
$S = (S \cap L) \cup (S \cap R)$
Let $S_L S \cap L$ and $S_R = \cap R$.
Consider the bipartite subgraph $G_1 = (V_1, E_1)$ of G induced on the vertex bipartition (i.e. with vertex partition):
$V_1 = S_L \cup (R - S_R)$.



$G_1 = (V_1, E_1) \qquad V_1 = S_L \cup (R - S_R)$

Suppose that W is any subset of S_L.
Let $N_{G_1}(W)$ denote the neighbor set of W of vertices in G_1.
In other words, $N_{G_1}(W) \subseteq R - S_R$.
We are going to show that $|W| \leq |N_{G1}(W)|$.
Suppose that $|W| > |N_{G1}(W)|$.
$\Rightarrow \exists w \in W$ such that $N_{G_1}(W - \{w\}) = N_{G_1}(W)$
$\Rightarrow (S_L - \{w\} \cup S_R = S$ is a vertex cover of G, contradicting the fact that S is a minimum vertex cover.
So $|W| \leq |N_{G_1}(W)|$.
By Hall's theorem, since G_1 satisfies Hall's condition, G_1 has an S_L-saturating matching M_1, with $|M_1| = S_L|$.
Now, let $G_2 = (V_2, E_2)$ be a bipartite subgraph with vertex partition:
$V_2 = (L - S_L) \cup S_R$.
By an argument similar to the above, we have that G_2 has an S_R-saturating matching M_2, with $|M_2| = |S_R|$.
Note that $|M = M_1| \cup M_2$ is a matching in G and
$|M| = |M_1| + |M_2| = |S_L| + |S_R| = |S|$.
Since in general $|M| \leq |S|$ and here we exhibited a matching and a vertex cover of equal size, we have that M is a maximum matching and S is a minimum vertex cover. $\square$

The König-Egerváry theorem is just a different interpretation of König's theorem which involves 0/1 matrices.

Theorem 2.5.2 *The König-Egerváry theorem:*
Let A be a 0/1 matrix.
Then the maximum number of ones in A, no two of which lie in the same row or column, is equal to the minimum number of rows and columns that together contain all the ones in A.

Example:

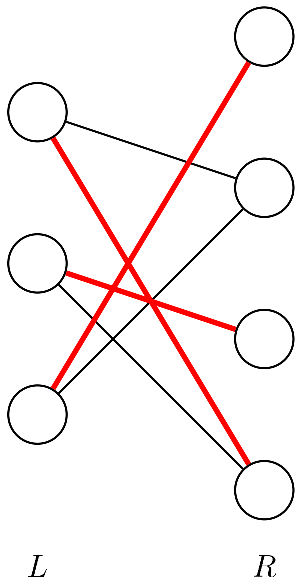
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \boxed{1} \\ 0 & 0 & \boxed{1} & 1 \\ \boxed{1} & 1 & 0 & 0 \end{bmatrix}$$

The ones in red are such that no two are in the same row or column.
There are three of them.
There are 3 rows and 4 columns of ones.
$min\{3, 4\} = 3$ (which is the same as the number of ones marked in red) □

As to the **proof** of the theorem, it consists of two steps.
Step one:
Given A, construct a bipartite graph $G = (V, E)$ with vertex partition $V = L \cup R$ and use A as an adjacency matrix of G, where L is the set of vertices corresponding to the rows of matrix A, and R is the vertex set corresponding to the columns.

Example:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \boxed{1} \\ 0 & 0 & \boxed{1} & 1 \\ \boxed{1} & 1 & 0 & 0 \end{bmatrix}$$



Step two:
Apply König's theorem.

Example:
In the above example, note that the matching marked in red in G corresponds to the three ones marked in red in A. And the three vertices of L is a minimum vertex cover. □

The König-Egerváry theorem has a few applications, such as in scheduling.

2.5.1 The Bottleneck Problem

Suppose that a manufacturing process consists of five operations that are to be performed simultaneously on five machines.

machine / operations	M1	M2	M3	M4	M5
op1	4	5	3	6	4
op2	5	6	2	3	5
op3	3	4	5	2	4
op4	4	8	3	2	7
op5	2	6	6	4	5

The time in minutes it takes each machine to perform each of the five operations is given in the table.
We would like to determine if it is possible to schedule all five operations so that the whole process is completed within 4 minutes.
Construct a matrix A whose a_{ij} entry is such that:

$$a_{ij} = \begin{cases} 1 & \text{if op}i \text{ takes at most 4 minutes on machine M}j \\ 0 & \text{otherwise} \end{cases}$$

A is a 5×5 matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & \boxed{1} \\ 0 & 0 & \boxed{1} & 1 & 0 \\ 1 & \boxed{1} & 0 & 1 & 1 \\ \boxed{1} & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & \boxed{1} & 0 \end{bmatrix}$$

It is possible to find 5 ones no two of which are in the same row or column.
They are marked in red.
Therefore, it is possible to complete the whole process within 4 minutes.
As a matter of fact, the five ones give the assignments.

	duration
op1 is assigned to $M5$:	4 minutes
op2 is assigned to $M3$:	2 minutes
op3 is assigned to $M2$:	4 minutes
op4 is assigned to $M1$:	4 minutes
op5 is assigned to $M4$:	4 minutes

Since the operations are performed in parallel, all 5 will finish within 4 minutes. □

2.6 Weighted matching

We are given a graph $G = (V, E)$ in which each edge $e \in E$ has a weight $w(e) = w_{ij}$ where $e = (v_i, v_j)$ and $w_{ij} \geq 0$.
The problem consists in finding a matching of G with the largest possible sum of weights.
Note that the previous (unweighted) graph matching problem is a special case of the weighted matching problem; just let $w_{ij} = 1$ for all $(v_i, v_j) \in E$.
One can also assume that instead of G we have a complete graph letting the weights of the edges that are missing in G be equal to zero.
Similarly, we can assume that we have an even number of nodes. If it is not the case, just add a node with edges of weight zero incident to it.
So, a bipartite weighted graph can be assumed to be a complete bipartite graph with vertex partition $V = L \cup R$ where $|L| = |R|$.
Note that the solution with largest sum of weights will always be a complete matching.

It is sometimes more convenient to think of this problem as a minimization problem by considering costs c_{ij} where:
$c_{ij} = W - w_{ij}$ with $W = max\{w_{ij}\} + 1$.
Once again, tackling the weighted bipartite problem is easier than dealing with general weighted graphs.
The bipartite weighted matching problem is also known as the assignment problem, since it solves the best assignment of tasks to workers problem where the value of w_{ij} represents the "production" of the i^{th} worker when given task j to perform. The assignment problem can be described by a linear program. The Hungarian method solves the weighted matching problem for a complete bipartite graph with $2|V|$ nodes in $O(|V|^3)$ arithmetic operations.

Bibliography

[AMB93] Ravindra K. Ahuja, Thomas L. Magnanti, and Orlin James B. *Network Flows*. Prentice-Hall, Inc., 1993.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, 1990.

[Wes96] Douglas B. West. *Introducation to Graph Theory*. Prentice Hall, Inc., 1996.