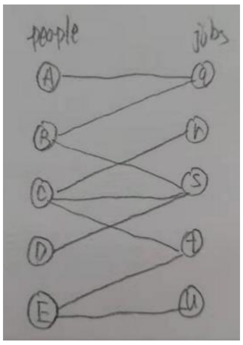
Part 7 Algorithms in Bipartite Graphs

(**Example 1**) Consider the following willing graph, where there are 5 vertices $\{A, B, C, D, E\}$ representing 5 people and 5 vertices $\{q, r, s, t, u\}$ representing 5 jobs. An edge represents a person is willing to do a job. There are some constraints on this willing graph:

(i) Each person can be assigned at most one job;

(ii) Each job can be assigned at most one person.

A natural question is that is it possible to assign a job to people such that (i) <u>every person gets a job</u> and (ii) <u>each job is filled</u>?



Consider the set of jobs that $\{A, B, D\}$ are willing to do, i.e., $\{q, s\}$, there's no **perfect matching** between the two sets. Consider the set of people willing to take up the jobs $\{r, t, u\}$, i.e., $\{C, E\}$, there's also no **perfect matching** between the two sets.

If the **perfect matching** doesn't exist, one can try to find the **maximum matching**.

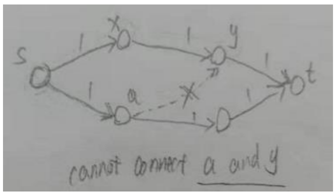
(**Definition 1**) Given a **bipartite graph**, we usually want to find the **maximum matching**. The problem of find the **maximum matching** on a **bipartite graph** can be formulated as the **max flow** problem, where we first convert a given bipartite graph to a corresponding **flow network**.

(1) Let L and R denote the sets of vertices on the left and right side of the bipartite graph. **Direct** all the edges of the bipartite graph from L to R.

(2) Add two new vertices s and t. Direct edges from s to all vertices of L and from all the vertices of R to t.

(3) Assign **capacity** 1 to all edges of the constructed **flow network**.

For simplicity, denote the original **bipartite graph** as G and the constructed **flow network** as G'.



(**Lemma 1**) (i) Suppose there is a **matching** with size k in G. Then, there is a **flow** with value k in G'.

(ii) Suppose there is a **flow** with value k in G', <u>where the flow on each edge is either 0 or 1</u>. Then, there's a **matching** with size k in G.

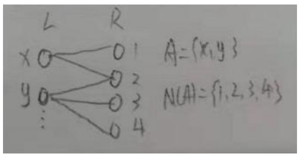
Thus, to find a **maximum matching** in G is equivalent to find the **max flow** in G'.

(**Remark 1**) In a flow network, in which all the capacities are integers, there's always a max flow that is integral (i.e., all the flow values $f(u, v)$ are integers). This follows from the **correctness** of the **F-F Method**.

Notes: Namely, when we run the **F-F method** on the given flow network, we can always obtain a max flow that is integral.

(**Definition 2**) Let (L, R) be the bipartite of a bipartite graph. For a subset $A \subseteq L$, define the set of **neighbors** of A as $N(A)$.

Notes: In the following example bipartite graph, let $A=\{x, y\}$. Then, we have $N(A)=\{1,2,3,4\}$.

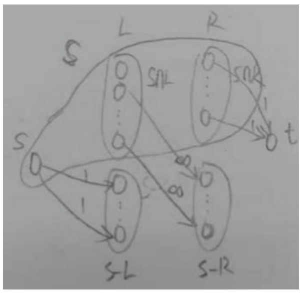


(**Theorem 1**) (**Hall's Theorem**) A **bipartite graph** $G=(V, E)$ with bipartite (L, R) has a **perfect matching** if and only if (i) $|L|=|R|$ and (ii) $|A|\leq|N(A)|$ for every $A \subseteq L$.

Proof of **Theorem 1**. (**Easy Direction**) If G has a **perfect matching**, then $|L|=|R|$ and $|A|\leq|N(A)|$ for every $A \subseteq L$, because every vertex in A must match to a distinct vertex in $N(A)$.

(**Difficult Direction**) Suppose that G does not have a **perfect matching** and $|L|=|R|$. We will prove that there exist $A \subseteq L$ such that $|A|>|N(A)|$.

We use the same reduction method as before to reduce the **bipartite matching problem** to a **max flow problem**. Especially, we let the **capacities** of edges from L to R to be **infinite**. Consider the s-t cut S with **minimum capacity**. A general form of S is illustrated as follow:



Let $n=|L|=|R|$. Since we assume that G does not have a **perfect matching**, we know that the value of maximum flow $|f|<n$. By the **Max-Flow Min-Cut Theorem**, we known that the capacity of S satisfies $c(S)<n$.

The s-t cut S may consist of the edges (i) from s to S-L, (ii) from $S \cap L$ to S-R, and (iii) from $S \cap R$ to t. Since we let the capacities of edges from L to R be **infinites**, there should be no edges from $S \cap L$ to S-R. Hence, we have

$$c(S) = |S-L| + |S \cap R| < n.$$

Moreover, we also have

$$|L| = |S \cap L| + |S-L| = n.$$

Then, we can obtain

$$|S \cap R| < |S \cap L|.$$

Let $A = S \cap L$. Since we already known that there should be no edges from $S \cap L$ to S-R, we have $|N(A)| < |S \cap R|$. Then, we have $|N(A)| < |A|$, which complete the proof.

(**Theorem 2**) (**Variant of Hall's Theorem**) A bipartite graph G=(V, E) with bipartite (L, R) has a **matching** that 'covers' all the vertices of L if and only if for every $A \subseteq L$, we have $|A| \le |N(A)|$.

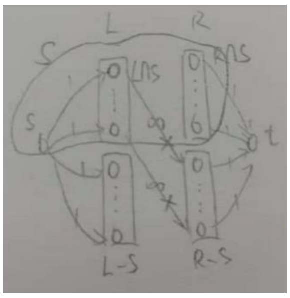
(**Algorithm 1**) We have the following algorithm to find a **minimum vertex cover** in a **bipartite graph** (where $|L|$ may not be equal to $|R|$).

(1) Find the **minimum cut** S, in the reduced flow network.

(2) Output $(L-S) \cup (R \cap S)$, denoted as C.

(**Claim 1**) **Algorithm 1** outputs a feasible **vertex cover** C.

Proof of **Claim 1**. For the **minimum cut** S found by step (1) of the algorithm, consider it general form as follow:



The capacity of S (i.e., $c(S)$) is the sum of the capacities of edges (1) between s and (L-S), (2) between $L \cap S$ and (R-S), and (3) between $R \cap S$ and t. Note that the $c(S)$ should be **finite**.

Since the capacities of edges between L and R are set to be **infinite**, there're no edges between $L \cap S$ and (R-S). Hence, the vertex set $C = (L-S) \cup (R \cap S)$ should cover all the edges in the original bipartite graph, i.e., edges between L and R.

(**Claim 2**) In the output of **Algorithm 1**, there is no vertex cover with size small than $|C|$, i.e., C is the **minimum vertex cover**.

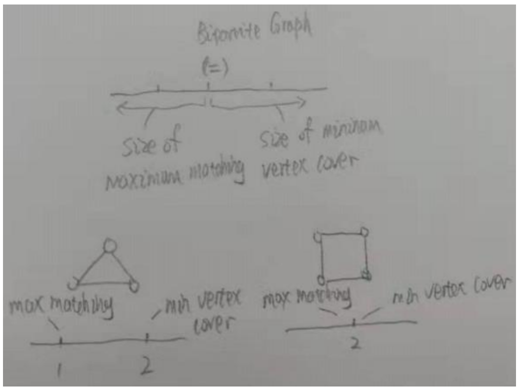
Proof of **Claim 2**. For the found **minimum cut** S, we have $|C| = c(S) = |L-S| + |R \cap S|$.

By the **Max-Flow Min-Cut Theorem**, we have that the value of maximum flow $|f|=c(S)=|C|$. Recall **Lemma 1** that in the reduced flow network of a bipartite graph G, the size of **maximum matching** of G equals the value of **maximum flow** $|f|$. Thus, we have the size of **maximum matching** of G equals $|C|$.

Recall that for any graph, the size of any **vertex cover** is at least (≥) the size of **maximum matching**. Hence, the size of any **vertex cover** is larger than or equal to (≥) $|C|$, which complete the proof.

(**Theorem 3**) (**Konig's Theorem**, 1931) In a **bipartite graph**, the size of **maximum matching** equals the size of **minimum vertex cover**.

Note: For a **general graph**, the size of **minimum vertex cover** is larger than or equal to (≥) the size of **maximum matching**.


Bipartite Graph
(=)
size of
maximum matching
size of minimum
vertex cover
max matching
min vertex
cover
1
2
max matching
min vertex cover
2