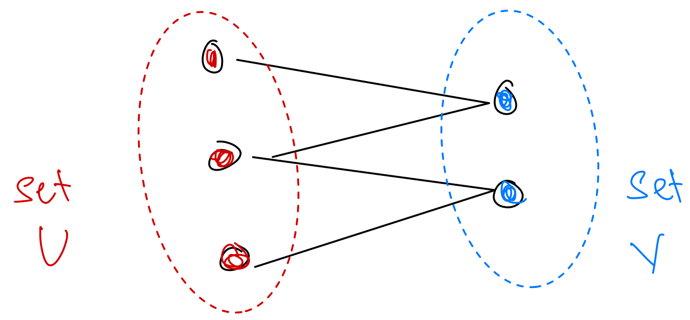
BIPARTITE ALGORITHM

Purpose:

Find maximal cardinality of bipartite graph $G = (U \cup V, E)$.
In other words, maximize the set of edges connecting 2 disjoint sets

Bipartite Graph:

Graph where vertices can be splitted into 2 disjoint sets.
There can't be no edges in the same group.



Methods to determine if a graph is bipartite:

- Detect cycle of odd length
- 2-colorable method:
  - Assign one color to a vertice
  - Assign a different color to its neighbor
  - Continue until you find 2 connected vertices with the same color

Hopcroft - Karp algorithm

An algorithm in the family of bipartite matching algorithms, which is used to find maximum number of matched pairs (maximum - cardinality) in a bipartite graph

Hopcroft - Karp pseudo

Input: Bipartite graph $G = (U \cup V, E)$
Output: Matching $M \subseteq E$

$M \leftarrow \emptyset$

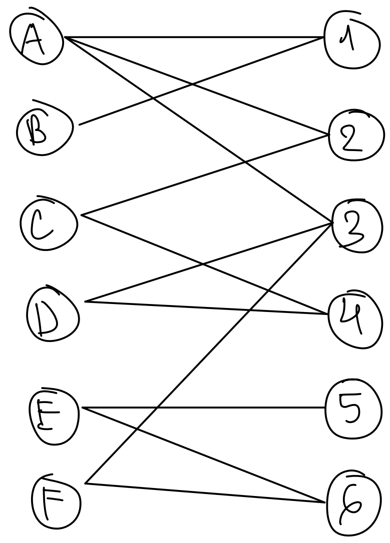
repeat:

- $P \leftarrow \{P_1, P_2, \ldots, P_k\}$ maximal set of vertex-disjoint shortest augmenting paths <BFS>
- $M \leftarrow M \oplus (P_1 \cup P_2 \cup \ldots P_k)$ <DFS>

until: there is no more augmenting path

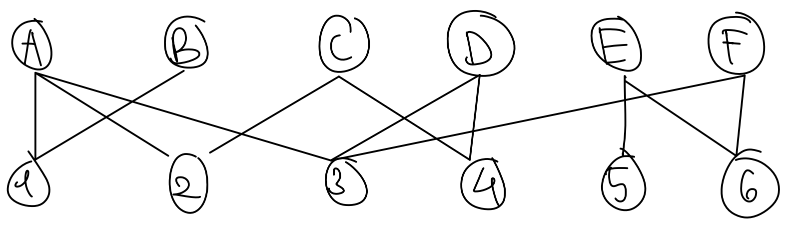
Hopcroft - Karp example:

Given this Bipartite graph, find maximal - cardinality



1st iteration:

Step 1: All vertices in left set are unmatched, choose all vertices to construct alternating graph:

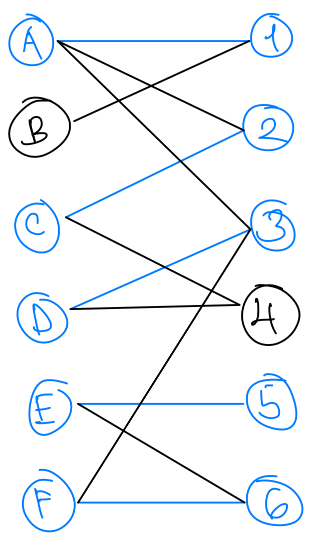


Step 2: Identify maximal set of vertex - disjoint shortest augmenting paths. In other words, the paths from unmatched leaves to roots.

$P = \{ A-1,\ C-2,\ D-3,\ E-5,\ F-6 \}$

The graph is now empty now that we remove all these paths and their vertices

Step 3: Augment original graph, update solution set



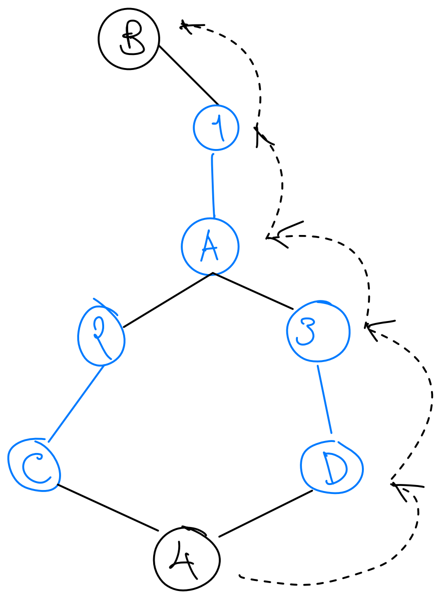
Note:

- anything in blue is "matched"
- anything in black is "umatched"

$M = \{ A-1,\ C-2,\ D-3,\ E-5,\ F-6 \}$

2^{nd} iteration:

Step 1: Construct alternating graph from umatched vertices



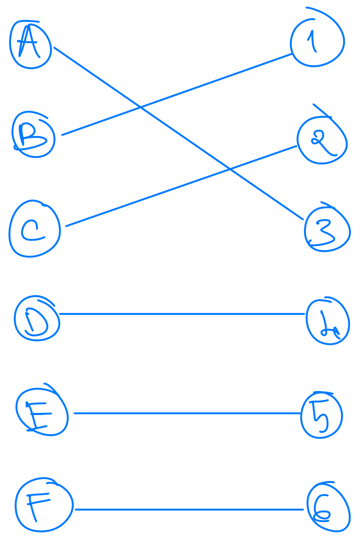
Step 2: Identify maximal set of vertex-disjoint paths

$P = \{ D\text{-}4,\ D\text{-}3,\ A\text{-}3,\ A\text{-}1,\ B\text{-}1 \}$

add in solution set

remove from solution set

Step 3:



$\Rightarrow M = \{ A\text{-}3,\ B\text{-}1,\ C\text{-}2,\ D\text{-}4,\ E\text{-}5,\ F\text{-}6 \}$

Since there is no more augmenting path, concluded that we reach maximal cardinality of provided graph.

Time Complexity:

Let $\begin{cases} |M^*| & \text{be the maximal matching} \\ |M| & \text{be the current matching} \end{cases}$

Lemma 1: After n iterations, the shortest augmenting path must be at least length n

Lemma 2:

$$|M| - |M^*| \leq |V|/n$$

where n is length of shortest augmenting path.

From Lemma 2:

$$|M| - |M^*| \leq |V|/n$$

From Lemma 1:

After $\sqrt{V}$ iterations, $n \geq \sqrt{V}$

Combined, we can say that:

$$|M| - |M^*| \leq \frac{|V|}{\sqrt{V}} = \sqrt{V} \quad (*)$$

In words, the number of remaining augmented paths is bounded by $\sqrt{V}$.

Connection to the time complexity, recall the pseudo code, $(*)$ means that "At iteration $\sqrt{V}$ iterations, we can only iterate a maximum of $\sqrt{V}$ times until there are no more augmenting paths" $\Rightarrow$ Maximum loops is $2\sqrt{V}$

Hopcroft - Karp pseudo

Input: Bipartite graph $G = (U \cup V, E)$

Output: Matching $M \subseteq E$

$M \leftarrow \emptyset$

repeat:

- $P \leftarrow \{P_1, P_2, \dots, P_k\}$ maximal set of vertex-disjoint shortest augmenting paths <BFS>
- $M \leftarrow M \oplus (P_1 \cup P_2 \cup \dots P_k)$ <DFS>

until: there is no more augmenting path

$\leq 2\sqrt{V}$ loops $\Rightarrow O(\sqrt{V})$

Now, inside each loop, we perform BFS then DFS, both are bounded by the number of edges in graph G, hence $O(|E|)$

So, the final time complexity is: $O(\sqrt{V}\,|E|)$