29 Aug 2025

Hopcroft – Karp Algorithm

Recap....

Naive Iterative Max Matching

NIMM(G):

Start with $M = \emptyset$.
while $\exists$ an M-aug path P
    $M \leftarrow M \oplus P$
endwhile
output M.

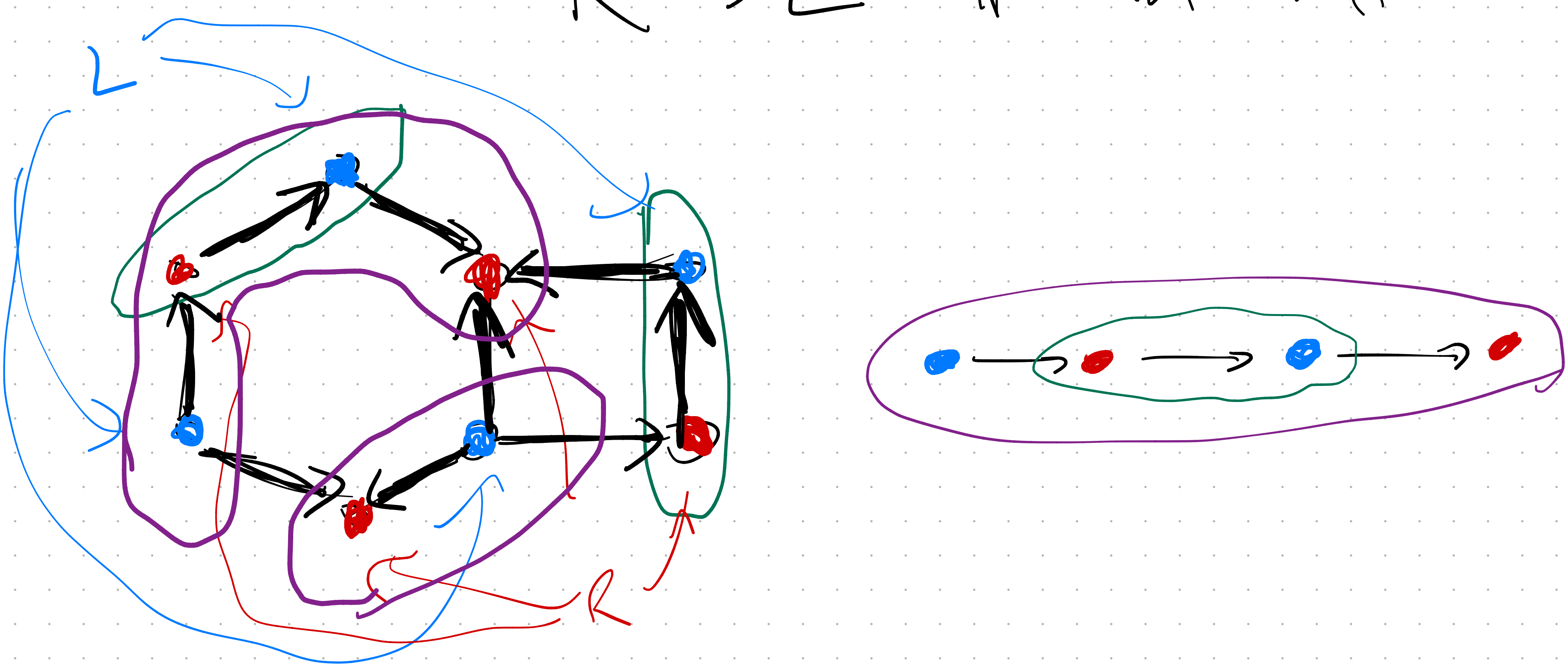
In bipartite graphs it is easy to find a M-augmenting path if one exists.

Def. A graph is bipartite if its vertices can be partitioned into two sets, L and R, s.t. every edge has endpoints in both sets.

Given bipartite G and matching M....

Def. Residual graph G_M is a directed graph with same vertices and edges as G, where edges oriented

- $L \rightarrow R$ if not in M
- $R \rightarrow L$ if in M.



$$\{M\text{-augmenting paths in } G\}$$

$$\Updownarrow$$

($F = \{\text{free vertices}\}$)

$$\{\text{directed paths in } G_M \text{ from } L \cap F \text{ to } R \cap F\}$$

BFS finds a path in this set, if one exists, in $O(m)$ time, where m = # of edges of G.

Naive Matching runs in $O(nm)$

where n = # vertices

m = # edges (assumed ≥ 1).

Hopcroft – Karp will improve this running time by finding disjoint augmenting paths at the same time.

<u>Lemma.</u> Let G be a graph, M a matching in G. Then following are equivalent for any $k \in \mathbb{N}$.

(i) there exists a matching M' with k more edges than M

(ii) there are k vertex-disjoint M-augmenting paths in G.

Def. If M is a matching in bipartite G,

let $d_M : V(G) \to \mathbb{N} \cup \{\infty\}$

be the function

$d_M(v) =$ # of edges in shortest M-alternating path from $L \cap F$ to v

or ∞ if no such path exists.

Equivalently, $d_M(v)$ is shortest path length in G_M from $L \cap F$ to v.

Def. An edge (u,v) of G_M is

<u>advancing</u> if $d_M(v) = d_M(u) + 1$.

otherwise <u>retreating</u>. $(d_M(v) < d_M(u))$.

An <u>advancing</u> augmenting <u>path</u>

is a path in G_M

from $L \cap F$ to $R \cap F$

made up of advancing edges.

<u>Hopcroft-Karp (G)</u>

Initialize $M = \emptyset$

while G_M has at least one

M-augmenting path

Let $\{P_1, \ldots, P_k\}$ be a

set-wise maximal set of

vertex disjoint advancing augmenting paths.

"Blocking set"

$M \leftarrow M \oplus (P_1 \cup \ldots \cup P_k)$

endwhile
output M.

How many times can the outer loop iterate?

Ans. At most $2\sqrt{n}$.

($\therefore$ running time of HK will be $O(m\sqrt{n})$.)

2 progress measures.

(a) Number of edges in M.

Grows by at least 1 per iteration.

(b) Length of shortest M-augmenting path.

Will prove: grows by at least 1.

Let M_t and M_{t+1} be the matchings in two consec. iterations.

If P is an advancing M_{t+1}-augmenting path, then P has at least one edge retreating w.r.t. M_t.

BFS tree w.r.t. G_{M_t}: