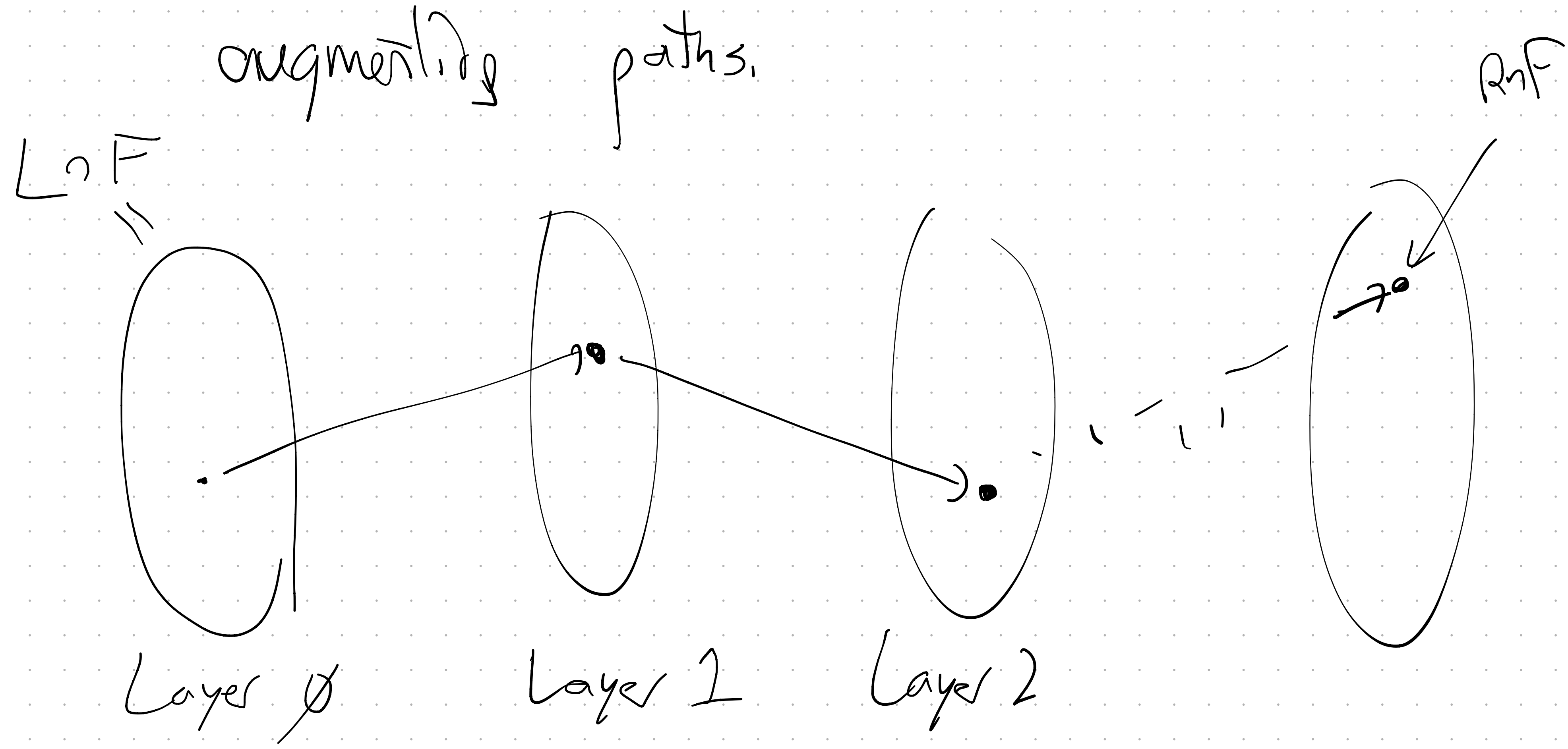
$\boxed{\text{3 Sep 2025}}$

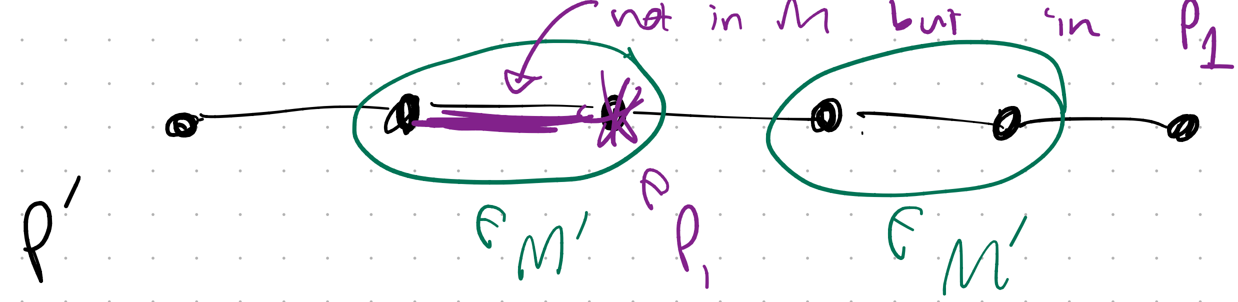
Finish Hopcroft — Karp.

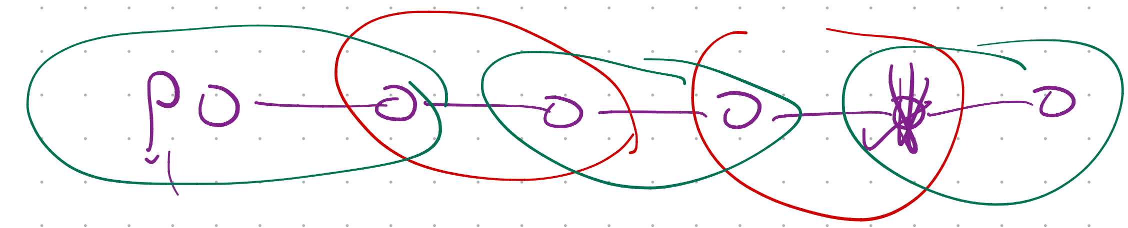
Introduce min-cost perfect matching.

<u>Def.</u> A blocking set is a setwise maximal collection of vtx-disjoint advancing augmenting paths.



<u>Lem.</u> If G is bipartite, M a matching, $Q = \{P_1, \dots, P_k\}$ is a blocking set w.r.t. M then the shortest augmenting path w.r.t. $M' = M \oplus (P_1 \oplus \dots \oplus P_k)$ is <u>strictly</u> <u>longer</u> than the shortest M-augmenting path.





<u>Upshot</u>: When we update M to

$$M \oplus (P_1 \oplus \cdots \oplus P_k)$$ using a

blocking set, two parameters

<u>strictly</u> increase:

1. # edges in matching
2. length of shortest M-augmenting path.

Running time analysis

After $\tau \triangleq \lfloor \sqrt{n} \rfloor$ iterations, shortest

M-augmenting path has length $> \tau$.

Suppose the max matching M^* has

k more edges than M_τ.

$\Longrightarrow$ $M^* \oplus M_\tau$ contains at least

k vertex-disjoint M_τ-augmenting

paths.

$\Longrightarrow$ In total those paths have $> k\tau$ vertices.

$\Rightarrow n > k\tau$

$$k < \frac{n}{\tau} = \frac{n}{\lfloor \sqrt{n} \rfloor}$$

$$k \leq O(\sqrt{n})$$

$\Rightarrow O(\sqrt{n})$ loop iterations remain.

Each iteration runs in $O(m)$

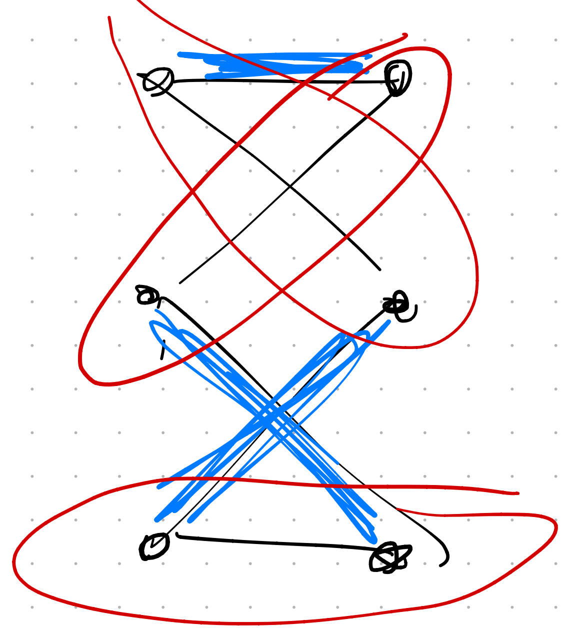
$\Rightarrow$ Hopcroft-Karp runs in

$$O(m\sqrt{n}).$$

<u>Min-cost bipartite perfect matching</u>

A <u>perfect matching</u> in a graph is one with no free vertices.

E.g.



MCBPM problem: given bipartite graph and cost $c(u,v)$ for each edge (u,v), find a perfect matching M that minimizes

$$c(M) \triangleq \sum_{(u,v) \in M} c(u,v).$$

Or, report no perf matching exists.