23 Aug 2023 — The Hopcroft–Karp Algorithm

Announcements

1. Waitlist
   - PINs not working for some courses. CS Dept aware of problem, trying to fix it. Please be patient.
   - Currently waitlist has 20 people, class has 14 open positions.
2. Overflow room: Gates G11.
   - Card key access required. CS PhD students have access
   - See Ed Discussion post for Zoom meeting code and permission code.
3. Homework 0: available on Gradescope. Will not be graded.
4. Prof K. office hrs canceled on Thurs, 8/24.
   Instead, extra office hour on Tues., 8/29, 5:30–6:30 pm.

"Naive" Bipartite Max Matching Alg.

Initialize $M = \emptyset$

While G contains an M-augmenting path P (alternating path whose endpoints are free vertices)
(found by building residual dir. graph G_M orienting M right→left and $E(G)\setminus M$ left→right
and running BFS):

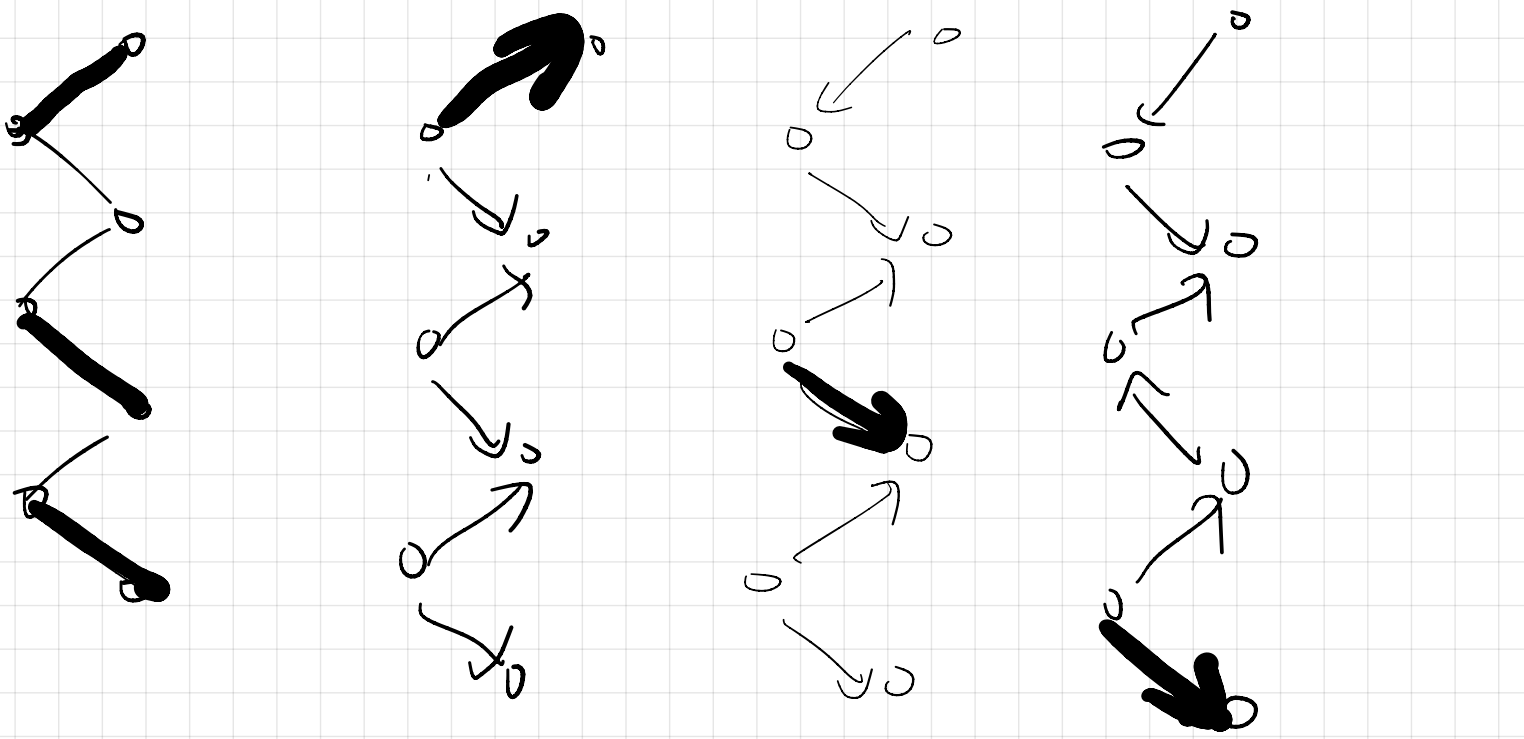
$M \leftarrow M \oplus P$

endwhile

output M

$\leq \frac{n}{2}$ loop iterations

$O(m)$ time per iteration (assuming no isolated vertices)



Hopcroft – Karp Algorithm

For bipartite graph G with matching M, let

L, R denote the sets of left & right vertices

F denote the set of vertices free with M

G_M denote residual graph of M in G.

If we do BFS of G_M starting from $L \cap F$ we can label each vertex v with the length of shortest path in G_M from $L \cap F$ to v. Call this $d(v)$.

$d(v) = 0$ if $v \in L \cap F$

$d(v) = 1$ if $v \in R$, and there's an edge from a free vertex to v

... and so on.

Edges of G_M, say (u, v), are either

"advancing": $d(v) = d(u) + 1$

or

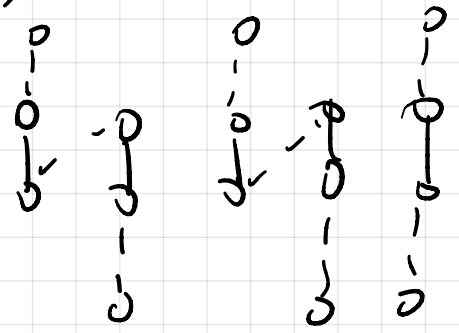
"retreating": $d(v) < d(u)$

Def. A blocking set of augmenting paths is a (setwise) maximal collection of vertex-disjoint advancing augmenting paths. (Those composed of advancing edges.)

bipartite

Fact 1. If G is a bipartite graph and M is any matching we can find a blocking set of augmenting paths in $O(m)$ time.

Fact 2. If M is a matching and $P_1, \ldots, P_k$ are vertex disjoint augmenting paths, $M \oplus (P_1 \cup P_2 \cup \ldots \cup P_k)$ is a matching with k more edges than M.



Fact 3. If M_0, M_1 are matchings and $|M_1| \geq |M_0| + k$ then $M_1 \oplus M_0$ contains at least k vertex-disjoint M_0-augmenting paths.

(Proof: repeat Monday's proof of the $k=1$ case, observe the graph $M_1 \oplus M_0$ must have k connected components each with one more edge in M_1 than in M_0. Each of those components is an augmenting path.)

H-K Algorithm

initialize $M = \emptyset$.

while G has an M-augmenting path:

let $P_1, \ldots, P_k$ be a blocking set of augmenting paths.

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

endwhile

output M