

$$\text{Let } \pi_i = \frac{\widehat{\pi}_i}{\sum_j \widehat{\pi}_j}$$

(Hitting time)

$$\pi_s = \frac{\widehat{\pi}_s}{\sum_j \widehat{\pi}_j} = \frac{1}{h_{ss}}$$

true for all s by uniqueness. \square

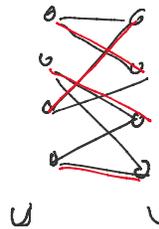
App'n: Perfect matching in Regular Bipartite Graphs

Given bipartite graph $G = (U \cup V, E)$, $|U| = |V| = n$, $|E| = m$.

classical algms for perfect matching:

$O(mn)$ time by augmenting paths

$O(m\sqrt{n})$ by Hopcroft, Karp



but faster algm possible for d -regular case (all vertices have deg d)

Gabow-Kariv '82: $O(m)$ time for $d = 2^k$.

⋮
 Cole-Ost-Schirra '01: $O(m)$ time for any d
 (related to edge coloring)

Surprisingly, can beat $O(m)$ using rand!!

Goel, Kapralov, Khanna '09: $\tilde{O}(n^{1.5})$ time
 good when $d \gg \sqrt{n}$
 * sublinear in input size

(by rand. sampling)

Goel. . . . '10: $O(n \log n)$ time -

Goal: ... 10: $O(n \log n)$ time -
by rand. walk!!

Review of Classical Algm:

$M = \emptyset$
repeat {

- find an augmenting path $v_0 v_1 v_2 \dots v_l$
s.t. $v_0 v_1 \notin M, v_1 v_2 \in M, v_2 v_3 \notin M,$
 $v_3 v_4 \in M, \dots, v_{l-1} v_l \notin M,$

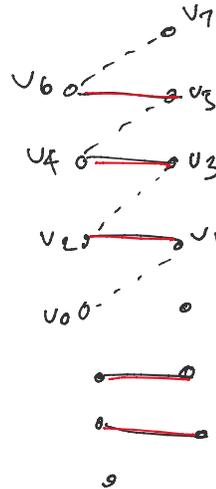
& v_0, v_l unmatched
 $v_i \in U, v_j \in V$

- delete $v_1 v_2, v_3 v_4, \dots$
& insert $v_0 v_1, v_2 v_3, \dots$ to M

\Rightarrow ($|M|$ increases by 1)

$\Rightarrow \leq n$ iterations

}



(correctness pf standard)

How to find aug. path:

Create dir graph H :

1. $\forall uv \in M (u \in U, v \in V),$
add edge (v, u)

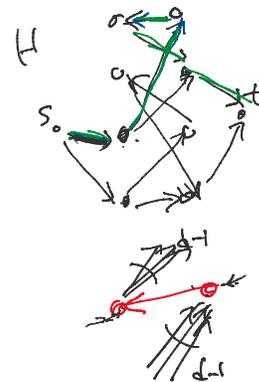
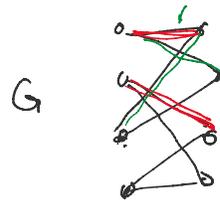
2. $\forall uv \in G - M,$
add edge (u, v)

3. add new vertices s, t
 $\forall u \in U$ unmatched, add (s, u)
 $\forall v \in V$ unmatched, add (v, t)

4. return a path from s to t in H
by BFS/DFS

$\Rightarrow O(m)$ time per iter.

$\Rightarrow \boxed{O(mn)}$ time



new idea - find path by rand. walk from s !

new idea - find path by rand. walk from s !

Lemma Given dir. graph H with m edges
 s.t. $\text{in-deg}(v) = \text{out-deg}(v) \forall v$

← (i.e. Eulerian graph)

Consider corresponding Markov chain,
 (with $P_{uv} = \frac{1}{\text{out-deg}(u)} \forall \text{edge } (u,v) \in H$)



Then $h_{vv} = \frac{m}{\text{out-deg}(v)}$.

Pf: Define $\pi_v = \frac{\text{out-deg}(v)}{m}$.

Check π is a stationary distrib:

$$\sum_v \pi_v = 1.$$

$$\begin{aligned} \forall v, \sum_u \pi_u P_{uv} &= \sum_{u: \text{in-neighbor of } v} \frac{\text{out-deg}(u)}{m} \cdot \frac{1}{\text{out-deg}(u)} \\ &= \text{in-deg}(v) \cdot \frac{1}{m} \\ &= \frac{\text{out-deg}(v)}{m} = \pi_v. \end{aligned}$$

Apply Fund. Thm. \square

issue - our H is not Eulerian
 but can fix it!

1. Contract each $uv \in M$ ($\text{in-deg} = \text{out-deg} = d-1$)
 3. duplicate (s,u) d times ($\text{in-deg} = \text{out-deg} = d$)
 (v,t) " " " "
- & contract s & t ($\text{in-deg}(s) = \text{out-deg}(s) = d(n-|M|)$)

... wanted time to find avg path

$$\Rightarrow \text{expected time to find aug path} \\ \text{is } O(h_{ss}) = O\left(\frac{m}{\text{out-deg}(s)}\right) \\ = O\left(\frac{dn}{d(n-|M|)}\right) = O\left(\frac{n}{n-|M|}\right).$$

$$\Rightarrow \text{total expected time} \\ O\left(\frac{n}{n-0} + \frac{n}{n-1} + \frac{n}{n-2} + \dots + \frac{n}{1}\right) \\ = \boxed{O(n \log n)}.$$

Harmonic numbers

Appl'n 2: Connectivity in Undirected Graphs

Given undir. graph $G=(V,E)$, $s,t \in V$,
decide whether t is reachable from s .

Easy: $O(m)$ time by BFS/DFS
needs $O(n)$ space

Question: space complexity?
 $O(\log n)$ (in bits)?

by Savitch's Thm, $O(\log^2 n)$ space

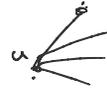
Aleliunas et al. '79: $O(\log n)$ space using rand!
("SL \subseteq RL ")

idea - just by rand walk

Lemma Let G be ^{connected} undir graph with m edges.
corresponding Markov chain $\text{edge} = 1$



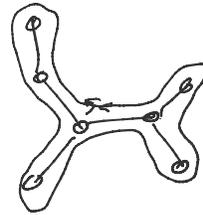
Lemma Let G be an undirected graph with n vertices and m edges.
 Consider corresponding Markov chain
 (with $P_{uv} = \frac{1}{\deg(u)} \forall u, v \in G$).



Then $h_{uv} \leq O(m)$ \forall edge $uv \in G$.

Cor $C_s =$ expected cover time
 $= E[\# \text{ steps to visit all vertices starting from } s]$
 $\leq O(mn)$.

Pf of Cor: Fix a walk s, v_1, v_2, \dots, v_k that visits all vertices of length $\leq 2n$
 e.g. by doubling a spanning tree

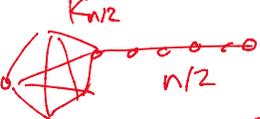


$$C_s \leq h_{s, v_1} + h_{v_1, v_2} + h_{v_2, v_3} + \dots + h_{v_k, v_k}$$

$$\leq 2n \cdot O(m) \quad \square$$

G : 
 Cover time = $\Theta(n^2)$

 K_n
 Cover time = $\Theta(n \ln n)$

 $K_{n/2}$
 $n/2$
 Cover time = $\Theta(n^3)$.