LONG PATH PROBLEMS

$G(V, E)$, integer k

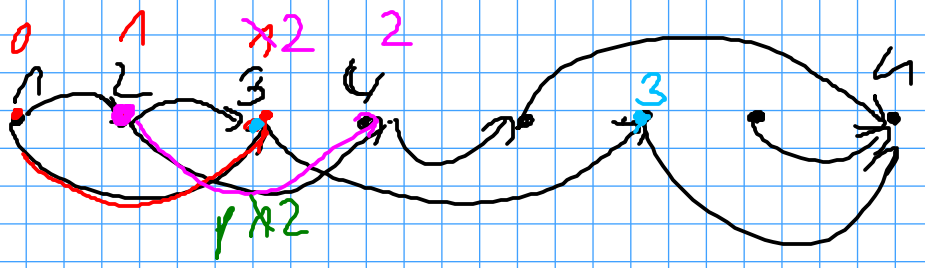
GOAL: find (simple) of length k in G

if $k = n-1$: Hamilton path problem NP-hard

$$k = O(\log n)$$

$$k = O\left(\frac{\log n}{\log \log n}\right)$$

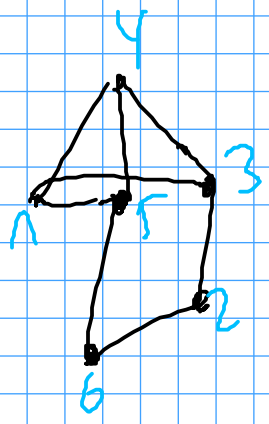
IDEA: Reduce the problem to find a long in a DAG (directed acyclic graph)

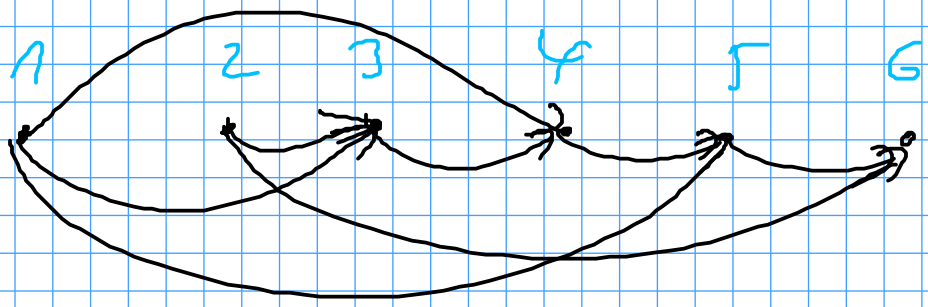


Topological sorting in $O(G)$

Runtime of sweep alg. to get max path label: $O(n+m+m) \in O(n \times m)$

Lemma: In a DAG, longest paths can be found in linear time.

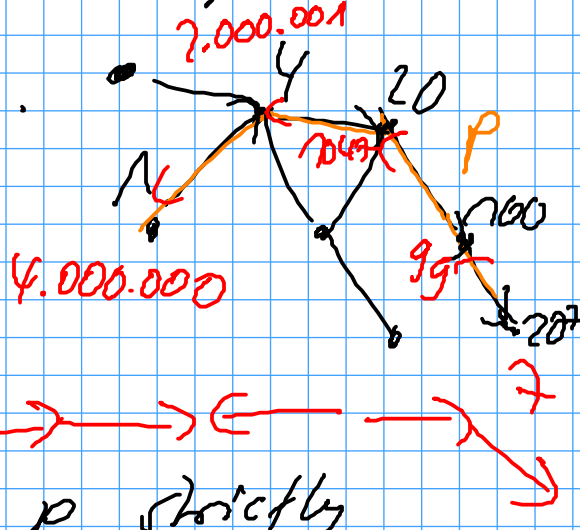




For some ordering of the nodes we can turn G in a DAG $\vec{G}$. But the order might influence the length of the longest path in the DAG.

Choose ordering u.a.r. (i.e. every permutation is equally likely.

Lemma: If G has a path p of length k, then p will be a directed path in $\vec{G}$ with probability $\frac{2}{(k+1)!}$.



Proof: From all possible permutations I only the ones where all labels for nodes on p strictly

increase or decrease, let p be a directed path in $\vec{G}$. Then only 2 out of $(k+1)!$ possible permutations count

Corollary: If G has a path of length k, then $\vec{G}$ has a path of length k with prob. at least α.

Algorithm:

Repeat $\frac{1}{\alpha} = t$ times

1. turn G in $\vec{G}$ (using a random node order) $O(t \cdot (n+m))$
2. find longest path in $\vec{G}$ using our linear time alg.
   - if we find a path of length k, we are done
   - else fail

$O(t \cdot (n+m))$

Theorem: This alg. finds a path of length

$$k = O\left(\frac{\log n}{\log \log n}\right) \quad \text{(if one exists)}$$

in random perfect graphs.

Proof: Assume there exists a path of length k in G.

$$P(\text{fail in all } t \text{ trials}) \leq (1-\alpha)^t \qquad 1+x \leq e^x$$

$$\leq e^{-\alpha t} \leq \frac{1}{e}$$

Thus the alg. finds a path of length k with prob. $\geq 1 - \frac{1}{e}$.

The running time is

$$O\left(\frac{(k+n)!}{2}(n+m)\right)$$

$$(k+n)! \leq (k+n)^q$$

$$O\left((k+n)^q (n+m)\right)$$

$$k = \frac{c \log n}{\log \log n}$$ for some constant c.

$O(n \cdot m^c)$

Improved Algorithm

Let C be a set of $k+1$ colours

Alg: color each vertex with a random colour from C

Def.: A colourful path in G is a path in which no colour repeats.

Obs: Longest colourful path in G is k long.

A colourful can be found in time linear in m but exponential in k.

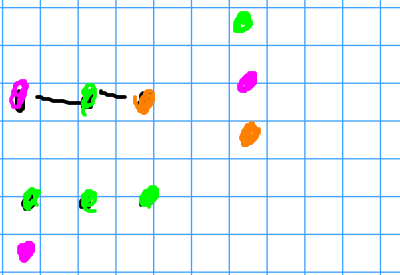
Exercise: Describe a dynamic programming algorithm which finds a colourful path of length k in $O(2^k m \cdot k)$.

Claim: If p is a path of length q in G then the probability that p is colourful is

$$\simeq \frac{\sqrt{2\pi(q+1)}}{e^{q+1}}$$

Proof: If we assign colors randomly to the vertices in p, the number of ways in which we can color p is

$$(q+1)^{q+1}.$$



The number of ways in which p is colorful is:

$$(q+1)!$$

$$\Pr(p \text{ is colourful}) = \frac{(q+1)!}{(q+1)^{q+1}}$$

(Stirling approximation:

$$n! \simeq \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$$

)

$$P(p \text{ is colorful}) \simeq \frac{\sqrt{2\pi(k+1)}\left(\frac{k+1}{e}\right)^{k+1}}{(k+1)^{(k+1)}}$$

$$= \frac{\sqrt{2\pi(k+1)}}{e^{k+1}} = \beta$$

Algorithm for long path:

Repeat $t = \frac{1}{\beta}$ times

1. Color the graph with $k+1$ colors randomly
2. find longest colorful path

Theorem: The alg. finds a path of length $k = O(\log n)$ in rand. polynomial time.

Proof: $P(\text{fail in all } t \text{ trials})$

$$\leq (1-\beta)^t$$

$$\leq e^{-\beta t} \leq \frac{1}{e}$$

Total runtime:

$$O\left(2^k m k \cdot e^{k+n}\right)$$

For $k = c \log n$:

$$O\left(2^{c \log n} m c \log n \, e^{c \log n + n}\right)$$

$$O\left(n^c m \log n^c\right)$$

$$O\left(n^{2c} m \log n\right) \in O\left(m n^{O(c)}\right)$$

$$e^{\left(\sqrt{(\log q / \log\log q)}\right)}$$