CSE596 Lecture Wed. Oct. 30, 2019

S-T Path Problem aka Graph Accessibility Problem (GAP)

INST: A directed graph $G=(V,E)$, and nodes $s, t \in V$. (source, sink)

QUES: Is there a path from s to t in G? — wlog G can be acyclic.

Hence NE_{NFA} and thus E_{NFA}, E_{DFA}, NE_{DFA} are all in P

GAP $\in$ P: We can do Breadth-First Search (BFS) from s to grow the set B of nodes reachable from s in $O(|E|)$ time given "random access" to generate neighbors of nodes.

When BFS closes, answer is yes $\iff$ t is in the final B.

$s=1$ Read Only

$\boxed{\Delta\,(1,2)(1,5)\cdots(i,j)\cdot(5,8)\cdot\ (27,29)}$

In fact GAP $\in$ NL $\equiv$ NSPACE$[O(\log n)]$.

R_1 $\boxed{\cancel{1}\,5\,8}$ label of current state, initially $s=1$ $\quad \boxed{s_2}$ R_1'

R_2 $\boxed{29}$ label of target state t $\quad \boxed{t_2}$ R_{t_2}

$\underbrace{\qquad\qquad}_{O(\log n)\text{ wide}}$

N uses only logspace.

$R_1 = s$

while $(R_1 \neq t)$:

N: Traverse the read-only input tape to guess an out-neighbor j of R_1

Overwrite R_1 by j

If $R_1 == t$ accept;

Nondet^c If there is a path from s to t, then some computation by N will find it, so $\langle G,s,t\rangle \in L(N)$. Else, N cannot accept. $\therefore L(N) = $ GAP.

Friday we will see NL $\subseteq$ P by BFS in a graph built from possible IDs of N.

How about "GST_2" or "GAP2": Given nodes s_1, s_2 and t_1, t_2, are there paths from s_1 to t_1 and from s_2 to t_2? — that have no vertices in common?

DST_2 "Disjoint S-T Paths".

GST_2 is also in NL $\subseteq$ P

No longer clear to see in NL because N "forgets" previously visited nodes on the first path, say. If N stores visited nodes in the first path, it uses $O(n)$ space, not $O(\log n)$. If both paths were required to go in order of increasing labels, then OK?

In NL again, so in P.

This works even if we have a variable number $s_1, \dots, s_r$ and $t_1, \dots, t_r$ of source and target nodes for the disjoint increasing connecting paths.

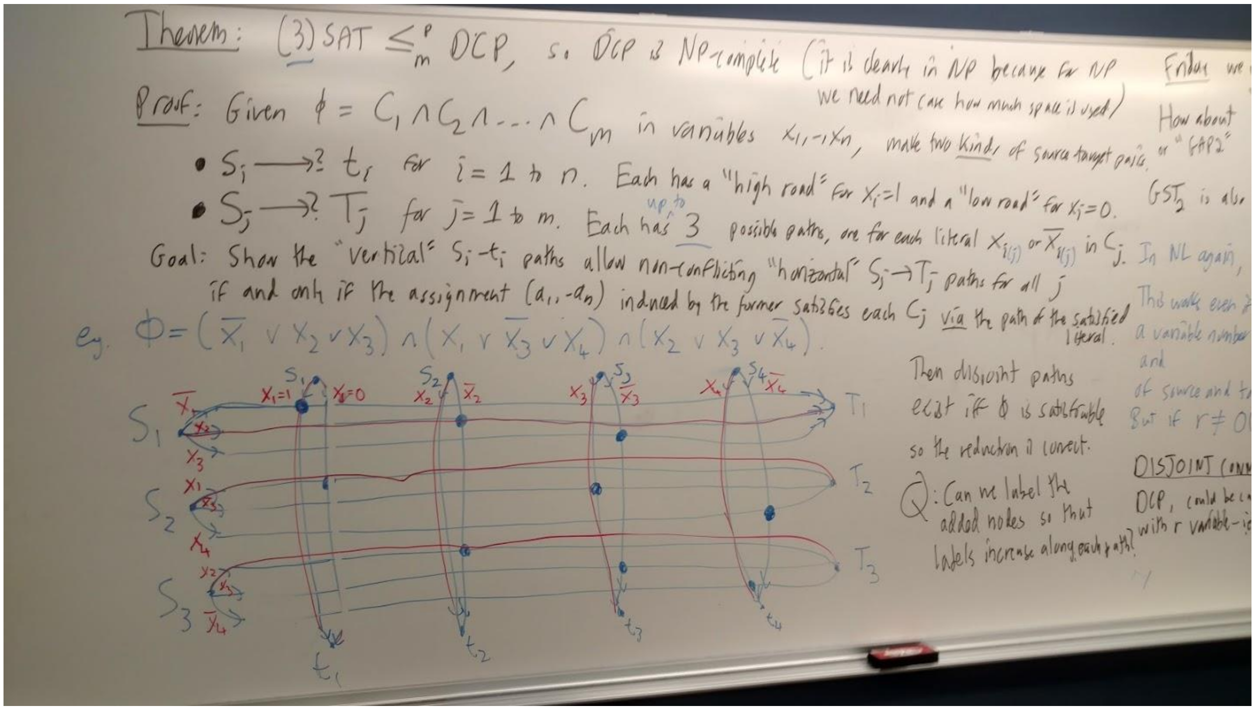
But if $r \neq O(1)$, this is more than $O(\log n)$ space. But in general:

DISJOINT CONNECTING PATHS

DCP, could be called "GST_r" with r variable—ie can depend on G.

INST: Directed G, $S=(s_1, \dots, s_r)$, $T=(t_1, \dots, t_r)$.

QUES: Are there paths s_1 to $t_1, \dots, s_r$ to t_r such that no two paths share a vertex?



The answer to the final question, as noted on Piazza by one of our undergraduate visitors, is *yes*: we simply use a "row-major" (or "column-major" or one that fans out from northwest to southeast) indexing which will be increasing down and to the right everywhere. What this means is that the problem remains hard even under the restriction that the paths must be increasing with regard to the vertex labels. In that case, however, one needs to have a variable number of paths for the problem to be hard---if the r parameter is constant then the problem with that restriction belongs to NL as we saw. What we may explore further is the opposite possibility: if we have a fixed number r of paths but they can go up and down with regard to the node labels, how hard is the problem then?