

CS 624 - HW 7

Max Weiss

30 March 2020

- Running BFS from node u in the provided graph generates the following d and π data.

	r	s	t	u	v	w	x	y
d	4	3	1	0	5	2	1	1
π	s	w	u	\uparrow	r	t	u	u

- Consider the first node n assigned a given depth d by the BFS algorithm. This node gets assigned as successors in the resulting edge tree, all related node in the underlying graph which are as yet unexplored. More precisely, given a graph $G = (E, V)$, the BFS algorithm induces an edge tree E_π with the following property: at each depth, there exists a node n such that for all m with $(n, m) \in E$, if m has depth greater than n then $(n, m) \in E_\pi$.

Using this observation, a counterexample is easy to find. It suffices to arrange for there to be some depth at which no node has all unexplored related nodes as its related nodes. There must be at least two nodes at that depth, so that it must be greater than 0, but it can be 1:

	s	t	u	v	w
E_π	\uparrow	s	s	t	u
E	$[t, u]$	$[s, v, w]$	$[s, v, w]$	$[t, u]$	$[t, u]$

- Let W be the set of wrestlers, so that $|W| = n$, and let $R \subseteq W \times W$ be the set of matched-up pairs. I will describe an algorithm to decide whether there exists an X which satisfies

$$\forall u \forall v ((u, v) \in R \rightarrow (u \in X \leftrightarrow v \notin X)) \quad (1)$$

and to find an example if any exists.

Since there is no constraint on nodes unconnected by R , the problem can be split into subproblems each of which has an underlying set of wrestlers which is connected by R . So we can assume after all that W is connected by R .

Let $R' \subseteq R$ be a rooted tree on W such that every node in W is a node in R' . Let Z be the set of all nodes whose depth in R' has parity 0. I will argue that if any X satisfies (1), then Z satisfies (1).

Suppose that X satisfies (1), and further suppose that $(u, v) \in R$, so that $u \in X \leftrightarrow v \notin X$. Since R' is a tree, there is a unique simple path in R' from u to v . By induction on the length, follows from (1) that this path must contain an odd number of edges. However, the number of edges in the path is the sum of the depths of u and of v . Therefore, the depths of u, v must have different parities, so that $u \in Z \leftrightarrow v \notin Z$, as desired.

This observation suggests how to determine efficiently whether some such subset X exists. Using BFS, we first build a datastructure to implement, for the resulting breadth-first tree, a corresponding set Z of even-depth nodes, so that evaluation of membership in Z runs in constant time. We then run through each $(u, v) \in R$ and check whether $u \in Z \leftrightarrow v \in Z$. If this ever happens, then we simply halt with the answer “no”. Otherwise, having exhausted all of R we just return Z .

The first phase of the above procedure is just a stepwise elaboration of BFS by some extra constant-time operations. So it runs in $O(n + r)$ time, for $r = |R|$. The second phase requires at worst $r = |R|$ steps, each of which involves only constant-time operations, so its running time just $O(r)$. The running time of the whole procedure is then $O(n + r)$ as desired.

4. Here are the discovery, finishing and predecessor data for the DFS tree:

u	$u.d$	$u.f$	$u.\pi$
q	1	20	\uparrow
s	2	7	q
v	3	6	s
w	4	5	v
t	8	15	q
x	9	12	t
z	10	11	x
y	13	14	t
r	16	19	\uparrow
u	17	18	r

The tree edges are given by the pairs $u.\pi, u$ in the table above. Here is a classification of the nontree edges:

$w \rightarrow s$	back
$q \rightarrow w$	forward
$z \rightarrow x$	back
$y \rightarrow q$	back
$r \rightarrow y$	cross
$u \rightarrow y$	cross

5. Suppose that there is a path P from u to v , and that $u.d < v.d$. Must u be an ancestor of v in the resulting DFS tree? By the parenthesis theorem, we

must have that $u.f < v.d$. By the white path theorem, P must contain at least one node w such that $w.d < u.d$. Now, if every such w satisfied $w.f < v.d$, then there would have to be infinitely many of them, but the underlying graph is finite. So, there must be some w such that $w.d < u.d < u.f < v.d < v.f < w.f$, where w lies on the path from u to v .

An example of a graph which can host such a DFS search is one with an edge relation $\{(w, u), (u, w), (w, v)\}$

6. By the parenthesis theorem, if $u.d < v.d$ while u is not a DFS ancestor of v , then $u.f < v.d$. So, any solution to exercise (5) is a solution here.