BFS, DFS, and MST

Dr. Danny Nguyen

Transcribed by Thomas Cohn

9/6/2018

Defn: Breadth-First Search (or BFS) is an algorithm which produces a spanning tree of a connected graph.

Did we define it to be connected? Did we discuss the behavior of BFS on graphs which are not connected?

BFS starts out at a specific defined vertex R in the graph $G = (V, E)$, and begins building up "layers". $L_0 = \{R\}$, and $L_{i+1} = N(L_i) \setminus \bigcup_{j=0}^{i} L_j$. Each time we add a vertex w, which is a neighbor of vertex $v \in L_i$, to a layer L_{i+1}, we add the edge (v, w) to our tree. The algorithm ends when $L_d = \emptyset$.

Observations:

1. $\bigcup_{i=0}^{d-1} L_i = V$.
2. The tree obtained from BFS is not unique.
3. $x \in L_i \leftrightarrow d(x, R) = i$ (we could formally prove this with induction).
4. For every edge $(v, w) \in G$, one of the following is true for some i:
   (a) $v, w \in L_i$.
   (b) $v \in L_{i+1}$, $w \in L_i$.
   (c) $v \in L_i$, $w \in L_{i+1}$.

Proof of (4):
Because v and w are adjacent, $d(w, R) - 1 \leq d(v, R) \leq d(w, R) + 1$.
So if $v \in L_i$, then $d(v, R) = i$, so $d(w, R) \in \{i - 1, i, i + 1\}$. $\square$

Defn: Depth-First Search (or DFS) is an algorithm which produces a spanning tree of a connected graph.

Ditto the definition for BFS.

DFS starts out at a specific defined vertex R in the graph $G = (V, E)$. We also have a stack $S = \{R\}$, and are building up our tree T. While $S \neq \emptyset$, we look at the top item of S (denoted v). If $\exists w \in N(v) \setminus T$, we add (w, v) to T and push w onto S. If no such w exists, we remove v from S.

Defn: Given a rooted tree T with root R and distinct vertexes a and b, we say a is an ancestor of b if the unique path from b to R passes through a.

This is *my* working definition, since we never received a clear one. I'm happy to replace this with a better definition if somebody (or the textbook, which I don't have yet) has one.

Thm: For all edges $e = (a, b) \in G \setminus T$, a is an ancestor of b in T or vice-versa.
Proof: WOLOG assume the DFS algorithm sees a before b. Then $b \in T_a$, where T_a is the DFS subtree of a. $\square$

Another argument from picture.

Defn: An edge in a connected graph is called a bridge if its removal would make the graph not connected.

Thm: If $e = (a, b) \in T$ is not a bridge of G and a is an ancestor of b in T, then $\exists e' \in T$ connecting an ancestor of a to a descendant of b.
Proof: Proof by picture.

Defn: A Minimum Spanning Tree (or MST) is a spanning tree with the smallest possible total weight.

We can write the weight a function, where $w : E \to \mathbb{R}_{>0}$. Note that if w is a constant function, then any spanning tree is a minimum spanning tree, since every tree with n vertexes has $n - 1$ edges.

Defn: Kruskal's Algorithm is an algorithm to find the MST of a graph.

Kruskal's Algorithm starts with a forest $F = \{\{1\}, \{2\}, \dots, \{n\}\}$. While F is not connected, pick the smallest edge that connects two separate components, and add that edge to the tree. Also merge the two components it connected in F.

Proof of Kruskal's Algorithm: Let T_0 be the edges of the tree obtained from Kruskal's Algorithm, with the edges ordered such that $T_0 = \{e_1 \leq e_2 \leq \cdots \leq e_{n-1}\}$. Let $T = \{a_1 \leq a_2 \leq \cdots \leq a_{n-1}\}$ be a spanning tree. Our goal is to prove that $\sum_{i=1}^{n-1} w(e_i) \leq \sum_{i=1}^{n-1} w(a_i)$. We will prove the stronger claim that $w(e_i) \leq w(a_i)$ for every $1 \leq i \leq n - 1$.

We know that $w(e_1) \leq w(a_1)$. So assume that $w(e_k) > w(a_k)$ for some $1 < k \leq n - 1$.
Then $w(e_k) \geq w(a_k) \geq w(a_{k-1}) \geq \cdots \geq w(a_1)$, i.e., $w(e_k) \geq w(a_j)$ for all $1 \leq j \leq k$.

At step k of Kruskal's Algorithm, since it selected e_k, we know that $a_1, \dots, a_k$ weren't picked even though they're smaller than e_k, so they must each already be in a component. Let $H = \{a_k, \dots, a_1\} \subset F_{k-1}$; H has $n - k$ components. But there are $n - k + 1$ components to F_{k-1}. **How exactly is this a contradiction? How did we get here?** $\square$