## Analysis of breadth-first search

How long does breadth-first search take for a graph with vertex set V and edge set E? The answer is O(V+E) time.

Let's see what O(V+E) time means. Assume for the moment that  $|E| \ge |V|$ , which is the case for most graphs, especially those for which we run breadth-first search. Then  $|V| + |E| \le |E| + |E| = 2 \cdot |E|$ . Because we ignore constant factors in asymptotic notation, we see that when  $|E| \ge |V|$ , O(V+E) really means O(E). If, however, we have |E| < |V|, then  $|V| + |E| \le |V| + |V| = 2 \cdot |V|$ , and so O(V+E) really means O(V). We can put both cases together by saying that O(V+E) really means O(M+E). In general, if we have parameters X and Y, then O(X+Y) really means O(M+E) means O(M+E).

(Note, by the way, that a graph is **connected** if there is a path from every vertex to all other vertices. The minimum number of edges that a graph can have and still be connected is  $|\mathbf{V}| - \mathbf{1}$ . A graph in which  $|\mathbf{E}| = |\mathbf{V}| - \mathbf{1}$  is called a **free tree**.)

How is it that breadth-first search runs in O(V+E) time? It takes O(V) time to initialize the distance and predecessor for each vertex O(V) time, actually). Each vertex is visited at most one time, because only the first time that it is reached is its distance null, and so each vertex is enqueued at most one time. Since we examine the edges incident on a vertex only when we visit from it, each edge is examined at most twice, once for each of the vertices it's incident on. Thus, breadth-first search spends O(V+E) time visiting vertices.