

1 Spanning trees in connected graphs with per-vertex degree constraints

Assume that we have a graph that we wish to create a spanning tree for. This problem has been proven by Kruskal and other mathematicians, however in this case we wish to consider an additional constraint. Assume that in the graph every vertex has some degree n , where for a spanning tree to be correct that vertex must have a degree in the spanning tree exactly equal to that degree n

Theorem 1. *Given a strongly connected graph $G(E, V)$ with total vertex degree of $2(n - 1)$ if algorithm 1 returns a spanning tree it will be correct*

Proof. Assume our algorithm returns an incorrect spanning tree for G . Our algorithm must necessarily cover every edge, as all the edges will have connections equal to their degree, and no degree can be zero. Additionally our spanning tree will feature $n - 1$ edges, where n is the number of vertices, due to the total vertex degree constraint of $2(n - 1)$. Therefore, as every vertex is touched, and the total number of edges is equal to $n - 1$, the returned tree must be a spanning tree. \square

Theorem 2. *Given a strongly connected graph $G(E, V)$ with total vertex degree of $2(n - 1)$ algorithm 1 will return a spanning tree*

Proof. For our algorithm to not complete, a component of the graph X must exist where the total degree of X is 0, and X is not G . This case would by definition not be a spanning tree of G . However X could not occur, as for X to occur the minimum and maximum nonzero vertex in G , forming new edge e , would have to be in X , and they would therefore have to both have a degree of one. If this is the case then due to the total vertex degree constraint all edges in the graph other than those in e would have a degree of zero, and thus assuming there are no cycles $X \cup E$ would have to be a spanning tree of V .

Additionally, our algorithm would be unable to return a spanning tree if the algorithm created a cycle. For a cycle to occur, edges must create a path of some vertex v back to vertex v . Assume our algorithm is in progress, and assume graph P . Assume an incomplete spanning tree T with no cycles on P , currently with k edges. For a cycle to be created a new edge e would have to be drawn from $v1$ and $v2$, which both already are connected to the tree T . This would mean, WLOG, that $v1$ has the lowest nonzero degree in P and $v2$ has the highest. However this would be impossible, as for $v1$ and $v2$ to already be in T they would need to have previously been of either the lowest or highest degree in P . \square