Greedy Algorithm: Minimum Cost Spanning Tree

Mason U'Ren, James Musselman, Zander Nelson

October 21, 2016

Abstract

What is the largest number of edges that an n-grid may have?

Algorithm 1 MinCost Spanning Tree Algorithm

$A = \emptyset$ ▷ A: the MinCost Spanning Tree
for all $v \in G.V$ **do** ▷ G: the graph, V: vertex
    $makeSet(v)$
end for
for all do(u, v) in G.E sorted increasing by $weight(u, v)$
    if then$findSet(u) \neq findSet(v)$
        $A = A \cup (u, v)$
        $union(u, v)$
    end if
end for
return A

1 What is the largest number of edges that an n-grid may have?

Assuming that our n-grid is connected by a minimum cost spanning tree, then the maximum number of edges in the n-grid is $n^2 - 1$. An n-grid is defined as a graph of n^2 nodes which is organized as a square array of $n \times n$ points, which are then connected in the cardinal directions without creating a loop.

Take for example a n-grid where $n = 4$. If we were to connect every point to another, restricted to the allowable directions and neglecting if loops were created, we would have a maximum of 24 edges. By simple deduction we can say that the minimum cost spanning tree then must have less than 24 edges. Thus, after eliminating all the loops we can see that only $n^2 - 1 = 4^2 - 1 = 15$ edges remain.

This can be proven by simple induction as well.

Base Case: $n = 1$

Thus $n^2 - 1 = 1^2 - 1 = 0$ ✓

Inductive Step:
Assume that $n = k$ is true for $k^2 - 1$.
Let $n = k + 1$, then
$\iff (k+1)^2 - 1$
$\iff (k^2 + 2k + 1) - 1$
$\iff (k^2 - 1) + 2k + 1$

Group Notes: I'm not sure exactly how to solve this proof. Maybe you guys see it and can let me know how to finish it.