

Analysis of Greedy algorithm

1. Initialization (lines 1,2): $3n$ constant time operations, $O(n)$ time
 2. Sorting edges (line 3): n^2 edges⁴, comparison constant-time, so $O(n^2 \log n)$ time
 3. Processing edges (lines 4-10): for every edge, 2 lookups ($in[g]$ and $out[f]$, line 5) and 2 find-operations ($Conn(f)$, $Conn(g)$, line 5), 2 updates ($in[g]$ and $out[f]$, line 7) and 1 union-operation (line 8), and 1 more lookup (line 9, no. of components); so for each edge, 3 union/find operations and 5 constant-time operations (lookups, updates); altogether there are n^2 edges (not all are necessarily processed but may be); so in total at most $3n^2$ union/find operations and $5n^2$ constant-time operations = $O(n^2)$ time
 4. Return edges: $n - 1$ edges = $O(n)$ time
- Total time:** $O(n) + O(n^2 \log n) + O(n^2) + O(n) = O(n^2 \log n)$.

⁴In actual fact, there are $n(n - 1) \leq n^2$ edges, but the analysis is simpler with n^2 .