kCores(graph: (V, E), k)

Input: a maximal connected graph G that is simple. k is the number of cores

output: a maximal connected subgraph within which every vertex has at least k.

---

while bucket not empty:

Let b be bucket.

for all V in graph:

if deg(V) < k:

deg(V) = deg(V) - 1

for all neighbor of V:

deg(neighbor) -= 1

add vertex into bucket.

for all vertex in bucket$_i$

remove vertex from graph

return all vertex left in graph.

Get All KCores (G : (V, E)

---

input : A simple & connected graph

output : All the kcore found starting from zero

---

let k = 1

let core be kCores(G, k)

while core is not empty

k += 1

let core be kCores (G, k)

Proof:

Lemma: Suppose your algorithm returns the k-cores $X = [x_1, x_2, \ldots, x_{n-1}]$ and there exist optimal k-cores $OPT = [y_1, y_2 \ldots, y_{n-1}]$, where x_i, y_i are sets of vertices in i-cores. For all $i < n$, $y_i \subseteq x_i$

Basis: Suppose there is a graph G with V vertices and E edges s.t. deg(V) of all V is at least one. Then it must be true that all vertices in G belongs to 1-core where $i = 1$.

Hypothesis:

Suppose $1 \leq i \leq k$, then all vertices in Y_k is in $X_k = Y_{k-1}$ such that Y is a subset of X

Inductive step:

Let $i = k+1$ and let $v \in Y_{k+1}$

By definition of a k-core, all of $v \in Y_{k+1}$ must be in Y_k since a vertex belongs in Y_{k+1} if it has at least $k+1$ degree, meaning it must have k degree. By hypothesis, given Y_k is in $X_k = Y_{k-1}$, all vertices in Y_k is in X_k,

meaning Y_k is a subset of X_k.
For all v in X_k is available for pruning, but since deg (v) must be at least k ~~anyone~~ none of them will be pruned, making all of them exist in Y_k.

conclusion:

By PMI, the lemma is correct

Time complexity

It loops through all V until all E are removed,

so $O(|V| + |E|)$

loop all V

remove all edge

CPE 133 HW