Advanced Algorithm Analysis

Problem set - 2. (Monsoon 2022)

(CS1.301)

Name - MALLA SAILESH

Roll number - 2021101106.

---

3). Here we are said $d_1, d_2, \ldots d_n$ are natural number. so $d_i \geq 1$ for $1 \leq i \leq n$.

so the degree of any node is atleast (one).

NOTE :- It was said simple graph. so no multi edges and self loops.

→ i.e. not more than one edge exist b/w two vertices/nodes.

first, we will sort the list in descending order. ($d_1 \geq d_2 \ldots \geq d_n$)

let say $d_1 = k_1 - 1$ ($k_1 - 1 < n - 1$). If not no graph.

Now If there is a graph with degrees $d_1, d_2, \ldots d_n$ then there exist a graph with degrees $d_2 - 1, d_3 - 1, \ldots d_{k_1} - 1, d_{k_1+1}, \ldots d_n$.

we said that the node d_1 ~~with highest degree~~ has $(k_1 - 1)$ edges with next ~~k_1~~ highest degree vertices.

→ Repeat this process until all the nodes are done.

If any of the above values become negative then, there exist no graph. ~~If not there exist~~ ~~(or)~~ If not there ~~a graph.~~ ~~If~~ + exist a graph.

correctness :-

one way :- If there is a graph G with degree sequence ~~(d1, ..., dn)~~ $d_2 - 1, d_3 - 1, \ldots d_{k_1} - 1, d_{k_1+1}, \ldots d_n$

then there exists a graph G_1 with degree sequence $d_1, d_2 \ldots d_n$: by adding $(k_1 - 1)$ edges b/w d_1 and $d_2 - 1, d_3 - 1, \ldots d_{k_1} - 1$.

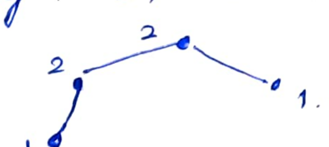
Repeat it for all nodes.

~~other~~ other way :- let say, there is a graph G with degree sequence $(d_1 \ldots d_n)$ where $d_1 = k_1 - 1$

so acc. to my algo i remove edges ~~b/w~~ ~~d_1~~ b/w d_1 and next $(k_1 - 1)$ highest degree vertices.

Proof by contradiction & counter example :-

let say; ~~we~~ there exist an edge b/w d_1, and $d_2, d_4 \ldots d_{k_1+1} \ldots$ When we repeat this for all nodes. Then there are ~~better~~ more chances that lower degree vertice becomes negative. eg :- 2, 2, 1, 1



counter example

2	2	1	1.
~~2~~	2	0	0
	~~2~~	-1	-1

we think graph ~~get~~ dont exist.

But exist

2 2 1 1.

we remove edges with highest degree vertices then it fine

↳ ~~2~~ 1 0 1

~~1~~ 0 0

so graph exist

Thus contradicts

which exist

There may be some other graph for
a list of natural number. but the given
algorithm works for all list of natural numbers.

① Given edge weights are distinct. First we will try to construct ~~an~~ a MST using Kruskal and union find algorithm.

Then we will take the value of edge e.

we know edge weights are distinct.

so we will check if the given edge weight is there in the weights of edges of MST constructed then given edge e is in the MST or else not there.

correctness:-

The key takeaway is edge weight are distinct. so there cant be two edges with same weight. so the existence of a particular edge e can be decided by the ~~checking~~ ~~that if~~ ~~weight~~ comparing the weight of edge e with weights of edges of MST.

②

we know M.S.T(T) of $G = (V, E)$ contains $v - 1$ edges. Now a new edge $e, (v, w)$ with edge weight c added to G.

→ we will calculate the weight of largest edge in MST (T)

case-1 : If given edge weight c is greater than ~~best case~~ largest weight edge in of (x) T (M.S.T)

then anyways this cant be included in M.S.T(T). so updated tree is also T

case-2 : If ~~a~~ given edge weight is less than largest weight of edge in T (M.S.T) ~~than~~. the edge ~~a~~ e, in same position in T also. Then we know that ~~will check~~. a cycle will be formed. because now v edges are there.

By using cycle property, we remove the largest weight in the cycle.

correctness : case-1

If we include that edge then by cycle property, [as anyways that weight of it larger than every edge weight] c is removed.

case-2 : we know we should only remove an edge from cycle.

Proof by contradiction,

If say some random edge is removed, from cycle then cycle property would fail. (This is infact what cycle property is about)

This contradicts.

so algorithm is correct

④ Given ($|X| = k$) terminals.

TO Prove:- problem of finding a minimum weight steiner tree on X can be solved in time $O\,n^{O(k)}$

Proof:- Let Y = extra nodes in Z (i.e which are not in X).

We first claim that each extra node has degree atleast 3 in T. If not by triangle inequality we "can" replace two incident edges by an edge joining its neighbours. so the other node gets isolated. so. let degree is atleast 3 ~~de~~ for extra node.

~~As there~~ ~~ate~~.

We know that every tree has atleast as many leaves as it has nodes of degree atleast 3.

$$|S| \leq \left(\begin{array}{c}\text{leave with degree} \\ \text{atleast 3}\end{array}\right)$$

$$|S| \leq K$$

Now if we compute M.S.T. on all sets of form $X \cup S$ with $|S| \leq k$, the cheapest among the all will be minimum steiner tree.

there are ${}^{n}C_{2k}$ such sets to try $\Rightarrow n^{O(k)}$.

Hence proved.

Referred. Kleinberg and Tardos and GFG for steiner tree.