10/24/16

ALGORITHMS

HW-6

Kiran Shettar

UML ID - 01605800

Problem 1 26.1-6 (pg 714)

It is obvious to see that the source is the projessors house & the sink or terminal is the school.

Since, the chidren are okay with crossing paths at a corner, we make every corner our vertex.

Ij there exists a path between any two corners, we draw an edge there. Since, only one can walk in a path and the other cannot, we set the edge capacity to be 1.

The net jlow at sink should be 2. Since both have to reach the school.

We just jollow the net jlow rule to get our solution $(j(u,v) \rightarrow 2)$ where 'j' is a net jlow junction.

Problem 3 26-5 (pg 762-763)

(a) The capacity of a cut is the sum of capacities of the edges that cross them.

The number of edges is at most $|E|$ & capacity of each edge is atmost C. Hence

The capacity of any cut G is atmost $c|E|$.

(b) Here we look for an argumenting ~~edg~~ path ~~and it's~~ where all the edges have capacity atleast 'k'. Then we have to do BFS or DFS so that we can find the path. We should not consider the lower capacity edges.

This type of search takes place in $O(V+E) = O(E)$ time.

Here, $|V| = O(E)$ in a flow network

(c) MAX-FLOW-BY-SCALING uses the FORD-FULKERSON-METHOD. Then it repeatedly checks until there is no path of capacity greater than or equal to 1 [i.e ≥ 1]. As all the capacities are integers & the path is always positive, means that there are no paths in the remaining path.

Hence, by MAX-FLOW MIN-CUT theorem, MAX-flow-BY-SCALING returns a maximum flow.

Problem 2 26.2.13 (pg 731)

Here, basically we have to construct a new graph G' with the same set of vertices & edges. Each capacity of the edge have to be increased by constant 'c'. By doing this the minimum cut will be increased & there will be least number of edges. That time the constant 'c' cannot be too large either. Because it might make cuts which was not minimum but with much fewer edges becoming minimum cut in the new graph.

We can define: $c = \frac{m}{2|E|}$

$m \rightarrow$ minimum difference b/n minimum cut & non-minimum cut in 'G'.

Then we increase every edge by constant 'c' then we have a new flow network G' in which any minimum cut in G' is a minimum cut with the smallest number of edges in G.