Red Scare! Report

by Group C.

Results

The following table gives my results for all graphs of at least 500 vertices.

Instance name	n	A	F	M	N	S
bht	5757	false	0	?!	6	true
common-1-1000	1000	false	-1	?!	-1	false
common-1-1500	1500	false	-1	?!	-1	false
common-1-2000	2000	false	-1	?!	-1	false
common-1-2500	2500	false	1	?!	6	true
common-1-3000	3000	false	1	?!	6	true
common-1-3500	3500	false	1	?!	6	true
common-1-4000	4000	false	1	?!	6	true
common-1-4500	4500	true	1	?!	6	true
common-1-500	500	false	-1	?!	-1	false
common-1-5000	5000	true	1	?!	6	true
common-1-5757	5757	true	1	?!	6	true
common-2-1000	1000	true	1	?!	4	true
common-2-1500	1500	true	1	?!	4	true
common-2-2000	2000	true	1	?!	4	true
common-2-2500	2500	true	1	?!	4	true
common-2-3000	3000	true	1	?!	4	true
common-2-3500	3500	true	1	?!	4	true
common-2-4000	4000	true	1	?!	4	true
common-2-4500	4500	true	1	?!	4	true
common-2-500	500	true	1	?!	4	true
common-2-5000	5000	true	1	?!	4	true
common-2-5757	5757	true	1	?!	4	true
gnm-1000-1500-0	1000	false	1	?!	-1	true
gnm-1000-1500-1	1000	false	2	?!	-1	true
gnm-1000-2000-0	1000	false	0	?!	7	true
gnm-1000-2000-1	1000	false	2	?!	-1	true
gnm-2000-3000-0	2000	false	0	?!	8	true
gnm-2000-3000-1	2000	true	2	?!	-1	true
gnm-2000-4000-0	2000	false	0	?!	6	true
gnm-2000-4000-1	2000	false	0	?!	5	true
gnm-3000-4500-0	3000	false	0	?!	10	true
gnm-3000-4500-1	3000	false	2	?!	-1	true
gnm-3000-6000-0	3000	false	0	?!	6	true

gnm-3000-6000-1	3000	false	2	?!	6	true
gnm-4000-6000-0	4000	false	0	?!	7	true
gnm-4000-6000-1	4000	false	1	?!	15	true
gnm-4000-8000-0	4000	false	0	?!	5	true
gnm-4000-8000-1	4000	true	2	?!	6	true
gnm-5000-10000-0	5000	false	2	?!	5	true
gnm-5000-10000-1	5000	true	1	?!	5	true
gnm-5000-7500-0	5000	false	-1	?!	-1	false
gnm-5000-7500-1	5000	false	-1	?!	-1	false
grid-25-0	625	true	0	?!	324	true
grid-25-1	625	true	0	?!	123	true
grid-25-2	625	true	5	?!	-1	true
grid-50-0	2500	false	0	?!	1249	true
grid-50-1	2500	false	0	?!	521	true
grid-50-2	2500	false	11	?!	-1	true
increase-n500-1	500	true	2	27	1	?!
increase-n500-2	500	true	1	30	1	?!
increase-n500-3	500	true	1	26	1	?!
rusty-1-2000	2000	false	-1	?!	-1	false
rusty-1-2500	2500	false	-1	?!	-1	false
rusty-1-3000	3000	false	0	?!	14	true
rusty-1-3500	3500	false	0	?!	14	true
rusty-1-4000	4000	false	0	?!	13	true
rusty-1-4500	4500	false	0	?!	7	true
rusty-1-5000	5000	false	0	?!	7	true
rusty-1-5757	5757	false	0	?!	7	true
rusty-2-2000	2000	false	0	?!	5	true
rusty-2-2500	2500	false	0	?!	4	true
rusty-2-3000	3000	false	0	?!	4	true
rusty-2-3500	3500	false	0	?!	4	true
rusty-2-4000	4000	false	0	?!	4	true
rusty-2-4500	4500	false	0	?!	4	true
rusty-2-5000	5000	false	0	?!	4	true
rusty-2-5757	5757	false	0	?!	4	true
smallworld-30-0	900	false	0	?!	9	true
smallworld-30-1	900	true	1	?!	11	true
smallworld-40-0	1600	false	0	?!	8	true
smallworld-40-1	1600	true	1	?!	13	true
smallworld-50-0	2500	false	0	?!	3	true
smallworld-50-1	2500	true	2	?!	-1	true
wall-n-100	800	false	0	?!	1	false
wall-n-1000	8000	false	0	?!	1	false
wall-n-10000	80000	false	0	?!	1	false
wall-p-100	602	false	0	?!	1	true

wall-p-1000	6002	false	0	?!	1	true
wall-p-10000	60002	false	0	?!	1	true
wall-z-100	701	false	0	?!	1	false
wall-z-1000	7001	false	0	?!	1	false
wall-z-10000	70001	false	0	?!	1	false

The columns are for the problems Alternate, Few, Many, None, and Some. The table entries either give the answer, or for those cases where there is a reason for our inability to find a good algorithm (because the problem is hard), we wrote '?!'.

For the complete table of all results, see the tab-separated text file `results.txt`.

Methods

None

For problem N, we solved each instance G by removing all red vertices from G and computing shortest path from *start* to *finish* by using Dijkstra's algorithm. The running time of this algorithm is $O(N + N^2)$ where N is a number of vertices. Our implementation spends 0.0258 seconds on the instance common-1-5000.txt with $n = 5000$.

Some

We solved the "some" problem for undirected graphs using the maximum flow (Ford-Fulkerson) algorithm. The graph was adapted by transforming it into a directed graph, then assigning capacity 1 to each vertex (by transforming them into 2 vertices one with all incoming edges one with all outgoing ones and connecting them with a single directed edge with capacity 1). Additionally, an extra start vertex was added and connected to the original start and finish with edges of capacity 1. Lastly, a new finish vertex was added, connected to one of the red vertices with an edge of capacity 2). If the flow found was of value 2, the algorithm found the path, if not, it would switch the edge between the sink and a red vertex to a next red vertex and repeat, until a path was found or all the red vertices were checked. The "some" problem cannot be easily solved for directed graphs and burden of proof is left as an exercise for the reader.

Many

Many can be solved by adding weight of one to all edges going to red vertices and all other edges with a weight of 0. Then finding the longest possible path. Unfortunately the longest path problem can only be solved for directed acyclic graphs in a polynomial time. For other graphs the longest path problem is NP and no known polynomial algorithm exists. This can be proven by reducing the Hamiltonian path problem to Many problem. This can be done by making all vertices red and then checking if result of many is equal to the amount of vertices in the graph. If so, then a Hamiltonian path exists through the graph.

Few

For problem few, we added weights of 1 to all edges going to red vertices and marked all other edges with weight 0. Then we used the Dijkstra's shortest path algorithm to find the shortest path and count the amount of red vertices in that path.

Alternate

For problem A, we solved each instance G by removing all edges that are connected with vertices that are both red or black and and computing shortest path from $start$ to $finish$ by using Dijkstra's algorithm. The running time of this algorithm is $O(E + V^2)$ where E is number of edges and V is number of vertices. Our implementation spends 0.0232 seconds on the instance common-1-5000.txt with $n = 5000$.