Problem

Given an undirected weighted graph, divide the vertices into two non-empty sets, such that, each vertex belongs to exactly one set, and the sum of the weight of all the edges having its two ends in two different such sets is maximum, which will be termed as maximum cut.

Algorithm

At first, a solution has been constructed based on greedy/semi-greedy/randomized techniques.
In the greedy technique, the edge with maximum weight has been selected initially to separate its two end vertices in two different sets. Then while choosing vertices from the remaining set, always the vertex contributing the maximum cut has been included in the restricted candidate list.
In the semi-greedy technique, the Restricted Candidate List of edges and vertices has been constructed based on the value of alpha, where 0 <= alpha <= 1. So the vertices which can contribute higher than this threshold, are included in the RCL, and from there, one is chosen at random.
In the randomized technique, no RCL is maintained. Instead, any vertex from the remaining set is chosen at random and included in either of the sets based on where it can contribute more to the cut.
After a solution has been constructed based on the above techniques, a local search technique has been applied. In the local search, the first neighbor has been picked, which can perform better if included in the other set.
Finally, the above algorithm is run multiple times, and their maximum has been taken as the final answer.

Output

The following table summarizes the maximum cut values obtained running against 54 different inputs:

Problem			**Constructive algorithm**			**Local search**		**GRASP**		**Known Best Solution**
						Local-1		**GRASP-1**		
Name	**n**	**m**	**Randomized-1**	**Greedy-1**	**Semi-greedy-1**	**No. of iterations**	**Best value**	**No. of iterations**	**Best value**	
G1	800	19176	11451	11460	11448	157	11364	50	11491	12078
G2	800	19176	11448	11491	11460	155	11368	50	11491	12084
G3	800	19176	11444	11492	11458	154	11361	50	11492	12077

G4	800	19176	11459	11492	11483	158	11373	50	11500	
G5	800	19176	11463	11521	11490	156	11369	50	11521	
G6	800	19176	2001	2022	2022	165	1907	50	2032	
G7	800	19176	1836	1839	1857	163	1741	50	1857	
G8	800	19176	1842	1854	1839	162	1747	50	1871	
G9	800	19176	1882	1892	1899	162	1787	50	1899	
G10	800	19176	1828	1864	1865	163	1742	50	1867	
G11	800	1600	465	506	494	14	446	50	506	627
G12	800	1600	452	502	496	15	433	50	502	621
G13	800	1600	479	526	526	17	457	50	526	645
G14	800	4694	2967	2992	2986	47	2940	50	2992	3187
G15	800	4661	2941	2969	2972	48	2921	50	2972	3169
G16	800	4672	2949	2975	2978	47	2925	50	2978	3172
G17	800	4667	2946	2976	2972	49	2923	50	2976	
G18	800	4694	874	911	930	72	834	50	930	
G19	800	4661	794	836	825	74	751	50	836	
G20	800	4672	812	880	849	71	779	50	880	
G21	800	4667	814	852	850	76	775	50	852	
G22	2000	19990	12895	13019	13002	239	12829	20	13019	14123
G23	2000	19990	12898	13014	13004	235	12828	20	13014	14129
G24	2000	19990	12883	13009	12988	239	12829	20	13009	14131
G25	2000	19990	12886	13029	13013	240	12826	20	13029	
G26	2000	19990	12882	13020	12981	244	12828	15	13020	
G27	2000	19990	2871	2979	2950	259	2820	15	2979	
G28	2000	19990	2842	2939	2892	256	2785	15	2939	
G29	2000	19990	2939	3037	3024	256	2881	15	3037	
G30	2000	19990	2942	3093	2991	259	2878	15	3093	
G31	2000	19990	2846	2921	2935	258	2793	10	2921	
G32	2000	4000	1128	1234	1224	39	1110	10	1234	1560
G33	2000	4000	1097	1222	1210	38	1082	10	1222	1537
G34	2000	4000	1091	1210	1216	41	1079	10	1210	1541
G35	2000	11778	7389	7450	7470	120	7368	10	7450	8000
G36	2000	11766	7376	7460	7450	120	7358	10	7460	7996
G37	2000	11785	7393	7487	7460	119	7374	10	7487	8009
G38	2000	11779	7386	7472	7449	120	7364	10	7472	

G39	2000	11778	2057	2157	2145	179	2016	10	2157	
G40	2000	11766	2031	2164	2120	191	1994	10	2164	
G41	2000	11785	2040	2163	2124	179	2004	10	2163	
G42	2000	11779	2114	2242	2186	185	2082	10	2242	
G43	1000	9990	6436	6500	6499	118	6399	10	6500	7027
G44	1000	9990	6433	6497	6477	120	6400	10	6497	7022
G45	1000	9990	6441	6501	6478	115	6399	10	6501	7020
G46	1000	9990	6434	6503	6482	118	6401	10	6503	
G47	1000	9990	6457	6514	6491	126	6410	10	6514	
G48	3000	6000	5134	6000	6000	70	5164	10	6000	6000
G49	3000	6000	5136	6000	6000	68	5162	10	6000	6000
G50	3000	6000	5149	5880	5880	68	5152	10	5880	5988
G51	1000	5909	3706	3753	3738	59	3691	10	3753	
G52	1000	5916	3710	3752	3753	58	3694	10	3752	
G53	1000	5914	3706	3743	3732	59	3691	10	3743	
G54	1000	5916	3706	3749	3743	58	3691	10	3749	

Here, n is the number of vertices in the graph and m is the number of edges.

In the Constructive Algorithm section, the column titled Randomized-1 represents the average performance of the randomized algorithm used to solve the given problem. In all the cases, 10 independent readings have been taken, and the floor of their arithmetic mean has been put down in this column.
The column titled Greedy-1 represents the performance of the greedy algorithm as described above.
The column titled Semi-greedy-1 represents the best performance out of different values obtained for different values of alpha in the semi-greedy algorithm as described above. The individual mappings of alpha and their corresponding output has been shown in the 1905045_Output_Semi-greedy.csv file.

Problem			**Constructive algorithm**		**Local search**		**GRASP**	
					Local-1		**GRASP-1**	
Name	**n**	**m**	**alpha**	**Semi-greedy-1**	**No. of iterations**	**Best value**	**No. of iterations**	**Best value**
G1	800	19176	0	11328	116	11369	50	11448

			0	11388				
			0.01	11297				
			0.03	11399				
			0.04	11361				
			0.09	11341				
			0.11	11448				
			0.14	11288				
			0.16	11315				
			0.17	11409				
			0.22	11348				
			0.25	11356				
			0.26	11305				
			0.31	11371				
			0.34	11348				
			0.35	11348				
			0.35	11360				
			0.35	11380				
			0.36	11317				
			0.36	11412				
			0.39	11331				
			0.42	11329				
			0.42	11373				
			0.42	11373				
			0.43	11389				
			0.52	11403				
			0.53	11396				
			0.54	11331				
			0.58	11373				
			0.61	11436				
			0.63	11389				
			0.65	11362				
			0.68	11397				
			0.69	11400				
			0.71	11376				
			0.72	11366				

			0.74	11390				
			0.75	11327				
			0.77	11353				
			0.8	11398				
			0.82	11428				
			0.84	11402				
			0.85	11440				
			0.87	11357				
			0.89	11367				
			0.9	11425				
			0.91	11364				
			0.92	11350				
			0.98	11385				
			0.99	11358				
G2	800	19176	0.02	11311	113	11381	50	11460
			0.02	11362				
			0.05	11358				
			0.07	11342				
			0.07	11379				
			0.12	11372				
			0.14	11353				
			0.15	11363				
			0.18	11438				
			0.19	11279				
			0.19	11444				
			0.2	11342				
			0.21	11341				
			0.21	11358				
			0.23	11370				
			0.28	11383				
			0.29	11381				
			0.35	11325				
			0.35	11411				
			0.5	11412				
			0.52	11436				

			0.55	11372				
			0.56	11359				
			0.56	11419				
			0.59	11354				
			0.61	11429				
			0.64	11397				
			0.64	11425				
			0.65	11344				
			0.65	11413				
			0.66	11444				
			0.68	11421				
			0.7	11419				
			0.71	11371				
			0.75	11362				
			0.76	11380				
			0.79	11422				
			0.8	11375				
			0.82	11338				
			0.82	11360				
			0.85	11360				
			0.87	11447				
			0.89	11341				
			0.9	11426				
			0.95	11397				
			0.96	11342				
			0.97	11382				
			0.97	11388				
			0.97	11460				
			0.99	11364				

In the Local Search section, No. of Iterations represents the average number of iterations performed during the local search for each GRASP iteration. And the Best Value represents the average best value obtained after each such local search.

In the GRASP section, No. of Iterations represents the total number of GRASP iterations run to obtain the maximum cut for that particular input, and the best value indicates the best result obtained out of all results of randomized, greedy and semi-greedy algorithms.

Here, for some inputs, it may seem that the best value has not occurred anywhere under the Randomized-1 or Greedy-1 or Semi-Greedy-1. Actually, it has come from the randomized run. But since only the average out of the 10 readings have been put down in the Randomized-1 column, the arithmetic mean came lower than the highest value.

Finally, in the Known Best Solution column, the benchmark values for 24 out of 54 inputs have been written to compare the actual upper bound vs our solution.

Discussion

In our generated outputs, the randomized, greedy, and semi-greedy solutions varied in almost all the cases. In some of the solutions, greedy came out as the best, whereas in some other solutions, semi-greedy or randomized algorithm performed better. It just confirms the fact that no algorithm is perfect for handling all types of inputs.

Again, for the semi-greedy algorithm, different alpha values produced different results. As the value of alpha increased, the size of the Restricted Candidate List of edges and vertices reduced. As a result, the algorithm got more and more inclined to make a greedier choice. And for the small values of alpha, the algorithm performed more like a randomized algorithm, because the formation of a Restricted Candidate List could not be that much significant.

In the Local Search phase, it can be seen that, the number of local search iterations were higher for larger and denser graphs, and smaller for smaller and sparser graphs. In small and sparse graphs, the choice of neighbour is few, so after few iterations, the local search algorithm reached the local maximum.

In the GRASP section, the choice of the number of iterations was independent. The result gets closer and closer to the upper bound as the number of iterations increases for randomized or semi-greedy algorithms. But due to lack of high quality hardware support and limited time constraints, the maximum number of iterations applied was 50. As a result, in some cases, the obtained answer was found to be a little less than the upper bound of the true solution.