You will be assigned one project from below. You will write a program to the given problem/algorithm. You will test with several input data. You will use C and compile and run on Liste.ctis.bilkent.edu.tr. We will test your program after compiling on liste machine, run test with supplied data and new data. Deadline will be Jan 9 2014 .

1. Prim -Dijkstra Algorithm for MST

   Given an undirected graph G=(V,E) V={1,2, .. n), E={1,2,..., m}. Graph will be given as n, m and list of m edges. Each edge will be given endpoints i and j for edge e=(i,j) and weight $w_e = w_{i,j}$.

   User will input root node: r

   Then algorithm will find a minimum spanning tree for root r, say T. T will be listed two ways: i) a list of edges, ii) a table showing parent pointers, with p(r)=-1. It will also output cost of T.

   You need to represent tree with parent points only. Compute and lists depth of the nodes in T.

2. Fundamental Cycles

   Given an undirected graph G=(V,E) V={1,2, .. n), E={1,2,..., m}. List all fundamental cycles with respect to T. Graph will be given as n, m and list of m edges. Each edge will be given endpoints i and j for edge e=(i,j) and $i < j$.

   User will input root node: r. Edge list will be given in two parts: T and $T^{\perp}$. You need to determine parent pointers of T with root r.

   The algorithm will output T as directed tree with root r via parent pointers p(i) and depth/height numbers h(i). You may use breadth first search among edges of T.

   Then algorithm will output for each edge $e \in T^{\perp}$, e and edges of C(T,e).

   Also number edges of T as 1, 2, n-1, and edges of $T^{\perp}$ as n, .., m. You need to form a matrix where rows are numbered 1i n-1 and columns are n, .. m and $C(i,j) = 1$ if $i \in C(T,j), =$ 0 otherwise .

   The program should output matrix C also.

3. MST Tree with Greedy (Kruskal's algorithm).

   Given an undirected graph G=(V,E) V={1,2, .. n} , E={1,2,..., m}. Graph will be given as n, m and list of m edges. Each edge will be given end endpoints i and j for edge e=(i,j) and weight $w_e = w_{i,j}$.

   We want to find a spanning tree whose total weight is maximum. You need to use heap to find edges with maximum weight and use union-find algorithm to check whether F+e contains a cycle. You start with Forest $F = (V, \emptyset)$ and stop with tree T containing all nodes or of size $|V| - 1$.

4. Shortest Path with Dijkstra Algorithm

   Given a directed graph with non-negative weights, and given r, you need to find a shortest path tree with d(i), p(i) for each node, where d(i) length of a shortest path from r to i, and p(i) parent of node i. Parent of root is set to -1. Nodes are numbered with positive integers.

   Input consists of:
   n number of nodes, for each node i, edges originating from i is given as
   i $j\ w_{i,j}\ k\ w_{i,k}\ ...\ -1$

   a node, say k, without any edges leaving it represented as
   k -1

   After inputting the graph , The program will prompt for root r and find the shortest path tree and at each iteration will print each node with d() and p(). At the end will print the the triples (i,d(i),p(i)) i=1, .., n.

5. Bellman-Ford Algorithm

   Given a directed graph with arbitrary integer weights, and given r, you need to find a shortest path tree with d(i), p(i) for each node, where d(i) length of a shortest path from r to i, and p(i) parent of node i. Parent of root is set to -1. Nodes are numbered with positive integers.

   Input consists of:
   n number of nodes, for each node i, edges originating from i is given as
   i $j\ w_{i,j}\ k\ w_{i,k}\ ...\ -1$

   a node, say k, without any edges leaving it represented as
   k -1

   After inputting the graph , The program will prompt for root r and You will start with d(r)=0 and $d(i) = \infty, i \neq r$, and $p(r) = -1, p(i) = 0$. You will apply Bellman-Ford Algorithm at most n iterations. You can stop earlier if d(i)'s become stable or d(r) becomes negative.

   The program should out triple (i, d(i), p(i)) i=1, 2, ..., n .

6. Network Simplex Method for Shortest path

   Given a directed graph with arbitrary integer weights, and given r, you need to find a shortest path tree with d(i), p(i) for each node, where d(i) length of a shortest path from r to i, and p(i) parent of node i. Nodes are numbered with positive integers.

   Input consists of:
   n number of nodes, for each node i, edges originating from i is given as
   i $j\ w_{i,j}\ k\ w_{i,k}\ ...\ -1$

   a node, say k, without any edges leaving it represented as
   k -1

   Let us add node numbered 0 as a dummy node, let r be real root. Let T_o be defined via d(r)=0, and $d(i) = 1000, i > 0, i \neq r$, d(0)=0, p(0)=-1, $p(i) = 0, i > 0$. Apply network simplex method until a) you find an an optimal solution to Shortest path problem rooted at node o or r, b) find a negative directed cycles.

   The Program should output, in a) triples (i, d(i), p(i)) in b) in addition give additional info for negative cycle.

7. Fundamental Co-Cycles

   Given an undirected graph G=(V,E) V={1,2, .. n), E={1,2,..., m}. List all fundamental cocycles with respect to T. Graph will be given as n, m and list of m edges. Each edge will be given endpoints i and j for edge e=(i,j) and $i < j$.

   Edge list will be given in two parts: T and $T^{\perp}$. root and parent pointers of T will be given as an array. parent pf node will be given as -1. all no numbers and edge numbers will be given as positive integers.

   Then algorithm will output for each edge $f \in T$, f and edges of D(T,e).

   Also number edges of T as 1, 2, n-1, and edges of $T^{\perp}$ as n, .., m. You need to form a matrix where rows are numbered 1, 2, ... n-1 and columns are n, .. m and $B(i, j) = 1$ if $i \in D(T, j), =$ 0 otherwise .

   The program should output matrix B also.

8. Topological Graphs

   Given Directed Graph G=(V,E) determine whether G is a directed graph. As output either you will give topological numbering, or output a directed cycle.

   You can choose any data structure for representing G.

9. General Knapsack Problem

   Given vectors, w, a and total resource b, give solution of problem

   $$\max\{wx \ : \ ax \leq b; x_j \in N\}$$

   You need to determine value of the objection function, solution vector x(b) and and algorithm which gives $x(t)$ for $t \leq b$.