CS2501

Data Structures

Graphs

For a single, global graph G, with vertices named by a unique non-negative integer, implement the following API to your graph data structure (the internal format/detail is up to you):

int adjacent(*x*, *y*): tests whether there is an edge from the vertices *x* to *y*; *return 1,0 (t/f)*

int[] neighbors(*x*): lists all vertices *y* such that there is an edge from the vertices *x* to *y*; *return an array of vertices, null if none*

int add\_vertex(*x*): adds the vertex *x*, if it is not there; *return 0 if it was already there, else 1*

int remove\_vertex(*x*): removes the vertex *x*, if it is there; *return 0 if it was not there, else 1*

int add\_edge(*x*, *y*): adds the edge from the vertices *x* to *y*, if it is not there; *return 0 if it was already there, else 1*

int remove\_edge(*x*, *y*): removes the edge from the vertices *x* to *y*, if it is there; *return 0 if it was not there, else 1*

void printEdges(): print a list of edges, one per line; print an edge by printing its two vertices

void printVertices(): print a list of vertices, one per line

This is essentially the list of basic graph operations from the Wikipedia page on Graphs.

Vertex – a unique integer