```
class mesh_t{
 private:
  // These variables are used to estimate the maximum.
  // number of bytes required to store an node or element.
  unsigned int max_len_node_CE;
  unsigned int max_len_node_fields;
  unsigned int max_len_elem_nodes;
  unsigned int max_len_elem_property;
  unsigned int max_len_elem_fields;

  // These values are stored so that we don't have to ask,
  // because a call to size() in deque can be expensive.
  unsigned int node_list_size;
  unsigned int element_list_size;
 public:

  // Data that defines a mesh. The reason a deque is choosen is because
  // there can be important resizes carried out. The meshes can get large
  // thus a realloc might wreak havoc on the computer.
  std::deque<node_t> node_list;
  std::deque<element_t> element_list;

  std::map< unsigned short, std::map<std::string, unsigned int> > scatter_nodes;
  std::map< unsigned short, std::map<std::string, unsigned int> > gather_nodes;

  // Constructer & distructer
  mesh_t();
  ~mesh_t();

  //
  // Methods.
  //
  void add_node(node_t&);
  void insert_node(node_t&, unsigned int pos);
  unsigned int MPI_max_node_size();

  void add_element(element_t&);
  void insert_element(element_t&, unsigned int pos);
  unsigned int MPI_max_element_size();

  // unn2gnn is used to index of a node that's stored
  // in node_list using the unn.
  unsigned int unn2node(unn_t unn);
  void unn2node_update();
  void unn2node_new();

  // The number of nodes that owned by this processor.
  // This is node_list.size() less the nodes that are
  // gatherwed from other processors
  unsigned int num_owned_nodes();

  unsigned int num_nodes();
  unsigned int num_elements();
  unsigned int num_scatter_nodes();
  unsigned int num_gather_nodes();

  void migrate();
 private:
  unsigned int __num_owned_nodes;
  std::map<std::string, unsigned int> unn2node_map;
};
```