Using **typedef:**

- Define data type of adjacency list
- Define data type of vertex using the **Vertex** class
- Define data type of subgraph using the **Subgraph** class
- Define data type of context of a task
- Define data type of task using the **Task** class

Need to trim adjacency lists?

YES

Define a subclass of **Trimmer**
{ Implement UDF trim(*v*) }

NO

Need an aggregator?

YES

Determine ValueT, PartialT, FinalT
Define a subclass of **Aggregator**
{ Determine what fields to maintain
Implement the 6 UDFs
}

NO

Define a subclass of **Comper**
{ Implement UDF task_spawn(*v*)
Implement UDF compute(*subg*, *context*, *frontier*)
}

Define a subclass of **Worker**
{ Write a constructor that calls **Worker**'s constructor
//specify number-of-threads, and paths for local files
Implement UDF toVertex(*line*)
Implement UDF task_spawn(*v*, *task_collector*)
}

Define the main() function
{ **init_worker(&argc, &argv);**
Create *param* and set its *param.input_path*
Create a worker object, pass in the number of compers
Create trimmer and aggregator objects (if applicable)
Bind them to the worker object (if applicable)
Call the worker object's run(*param*) function
worker_finalize();
}