

```
nifty::graph::optimization  
::lifted_multicut::LiftedMulticut  
ObjectiveBase< LiftedMulticutObjective  
< GRAPH, WEIGHT_TYPE >, GRAPH, UndirectedGraph  
<>, WEIGHT_TYPE >
```

```
std::vector< T >
```

```
nifty::graph::graph  
_maps::EdgeMap< G, T >
```

```
nifty::graph::optimization  
::lifted_multicut::LiftedMulticut  
Objective< GRAPH, WEIGHT_TYPE >
```

```
nifty::graph::Undirected  
GraphBase< UndirectedGraph  
< EDGE_INTERANL_TYPE, NODE  
_INTERNAL_TYPE >, detail_graph  
::SimpleGraphNodeIter, detail  
_graph::SimpleGraphEdgeIter, detail  
_graph::UndirectedGraphTypeHelper  
< EDGE_INTERANL_TYPE, NODE_INTERNAL  
_TYPE >::AdjacencyIter >
```

```
nifty::graph::Undirected  
Graph< EDGE_INTERANL_TYPE,  
NODE_INTERNAL_TYPE >
```

weights_

liftedGraph_