JGraphT

All Classes

Packages

org.jgrapht
org.jgrapht.alg
org.jgrapht.alg.clique

All Classes

AbstractBaseGraph
AbstractCapacitatedN
AbstractFundamentalC
AbstractGraph
AbstractGraphBuilder
AbstractGraphIterator
AHUForestIsomorphism
AhujaOrlinSharmaCap
AhujaOrlinSharmaCyc
AHURootedTreeIsomo
AHUUnrootedTreeIsom
AliasMethodSampler
AllDirectedPaths
AlphaCentrality
ALTAdmissibleHeurist
AlwaysEqualComparat
ArrayUnenforcedSet
ArrayUnenforcedSetE
AsGraphUnion
AsSubgraph
AsSynchronizedGraph
AsSynchronizedGraph
AStarAdmissibleHeur

iterator type.

Package org.jgrapht.alg.cycle Description

Algorithms related to graph cycles.

Algorithms for enumeration of simple cycles in graphs

Contains four different algorithms for the enumeration of simple cycles in directed graphs. The worst case time complexity of the algorithms is:

1. Szwarcfiter and Lauer - $O(V + EC)$
2. Tarjan - $O(VEC)$
3. Johnson - $O(((V + E)C)$
4. Tiernan - $O(V.const^V)$

where V is the number of vertices, E is the number of edges and C is the number of the simple cycles in the graph. All the above implementations work correctly with loops but not with multiple edges. Space complexity for all cases is $O(V + E)$.

The worst case performance is achieved for graphs with special structure, so on practical workloads an algorithm with higher worst case complexity may outperform an algorithm with lower worst case complexity. Note also that "administrative costs" of algorithms with better