Package ‘maxmatching’

October 13, 2022

Type Package

Title Maximum Matching for General Weighted Graph

Version 0.1.0

Author Bowen Deng

Maintainer Bowen Deng <baolidakai@gmail.com>

Description Computes the maximum matching for unweighted graph and maximum matching for (un)weighted bipartite graph efficiently.

License CC0

LazyData TRUE

Imports igraph

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-15 09:51:07

R topics documented:

`blossom` *Blossom’s algorithm*

Description

Computes the weighted bipartite matching using Hungarian’s algorithm

Usage

```
blossom(G, weighted = FALSE, maxcardinality = FALSE)
```

Arguments

`G` The graph to compute the maximum cardinality matching

`weighted` Whether the graph is weighted, if true, weights should be obtained by E(G)$weight

`maxcardinality` Whether the maximum weight should be computed over all maximum cardinality matchings

Details

Blossom's algorithm for maximum cardinality matching for general graph

Efficiently compute the maximum weighted biparitite matching using the Hungarian algorithm (TODO: citation) Almost a direct port of Joris van Rantwijk's python code at http://jorisvr.nl/files/graphmatching/20130407/m

Value

The maximum weighted matching for G, in a list of edges

`maxmatching` *Maximum Matching*

Description

Compute the maximum matching for undirected graph

Usage

```
maxmatching(G, weighted = FALSE, maxcardinality = FALSE)
```

Arguments

`G` undirected igraph object representing the input

`weighted` whether the graph is weighted, if the graph is weighted, the weight should be able to be accessed with igraph::E(G)$weight

`maxcardinality` Ignore if the graph is bipartite, and unmeaningful if the graph is unweighted. If the graph is non-bipartite and weighted, only computes the maximum weighted matching among all maximum cardinality matchings.

Details

maxmatching

TODO

Value

The matchings in a list

Examples

```
# Unweighted general graph
G1 <- igraph::graph(c(1, 2, 1, 3, 1, 4, 3, 4, 3, 5,
5, 6, 6, 7, 7, 8, 8, 9, 3, 8, 5, 8), directed = FALSE)
maxmatching(G1, weighted = FALSE)
# Unweighted bipartite graph
G2 <- igraph::graph(c(1, 5, 1, 6, 1, 7, 2, 5, 2, 8,
3, 6, 3, 7, 3, 8, 4, 6, 4, 7, 4, 8), directed = FALSE)
maxmatching(G2, weighted = FALSE)
# Weighted bipartite graph
G3 <- igraph::graph(c(1, 5, 1, 6, 1, 7, 2, 5, 2, 8,
3, 6, 3, 7, 3, 8, 4, 6, 4, 7, 4, 8), directed = FALSE)
igraph::E(G3)$weight <- runif(length(igraph::E(G3)), 0, 1)
maxmatching(G3, weighted = TRUE)
```

Index