Bipartite Maximum Cardinality Matching

ALGORITHM MATCHبNG [G = *(S, T; E), M, n]*

This algorithm determines a maximum cardinality matching M in a bipartite graph (
by augmenting a given matching in *G*.

INPUT: Bipartite graph G = *(S, T; E)* with vertices l, • • • , *n,* matching *M* in (
(for instance, *M* = 0)

OUTPUT: Maximum cardinality matching *M in G*

1. If there is no exposed vertex in *S* then
OUTPUT *M.* Stop
[M is of maximum cardinality in G.]
Else label ali *exposed* vertices *in S* with 0.
2. For each ; in *S* and edge (;', *j) not* in *M,* label *j* with ;', unless aiready labeled
3. For each *nonexposedj* in *T,* label ;' with *j,* where ; is the other end of the unique edge (;', *j)* in *M.*
4. Backtrack the altemating paths *P* ending on an exposed vertex in *T* by using the labeis on the vertices.
5. If no *P* in Step 4 is augmenting then
OUTPUT *M.* Stop
[M is of maximum cardinality in G.]
Else augment *M* by using an augmenting path *P.*
K^ilým/^ uu luü^la.

Go to Step l.
End MATCHبNG

EXAMPLE 1 Maximum cardinality matching

Is the matching Mý in Fi474 .ٌa of maximum cardinality? If not, augment it until maximum cardinalip
reached.

Solution. We appiy the algorithm.

1. Label l and 4 with 0.
2. Label 7 with l. Label 5, 6, 8 with 3.
3. Label 2 with 6, and 3 with 7.
[Ali vertices are now labeled as shown in Fi474 .ٌa.]
4. Pý: l — *~1* ~ 3— 5. *[By backtracking, P^ is augmenting.]*
Pg: l - 7 — 3 - 8. *[P y, is augmenting.}*
5. Augment Mý by using Pý, dropping (3, 7) from Mý and inciuding (1,7) and (3, 5). Remove ali labeli
Go to Step l.
Figure 474b shows the resulting matching Mg = {(l, 7), (2, 6), (3, 5)}.
1. Label 4 with 0.
2. Label 7 with 2. Label 6 and 8 with 3.
3. Label 1 with 7, and 2 with 6, and 3 with 5.
4. *ps: 5 —* 3 - 8. *[Py is altemating but not augmenting.]*

5. Stop. Mg *is of maximum cardinality (namety,* 3).