CS4234
Optimiz(s)ation Algorithms

L8 – (Weighted) Max-Cardinality -(Bipartite)-Matching

https://visualgo.net/en/matching

CS3233/CS4234
Dual Slides ☺

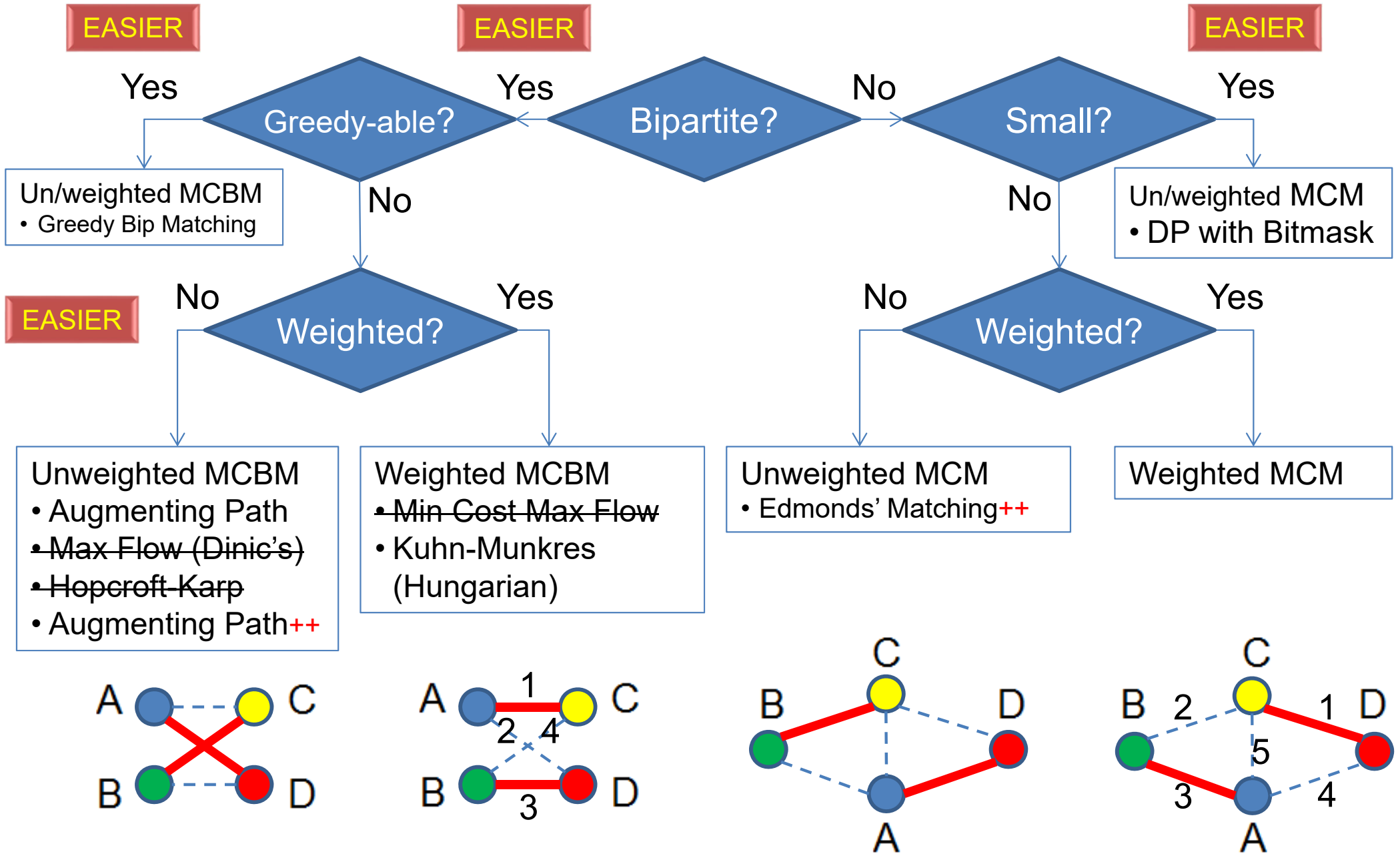
Dr. Steven Halim

Roadmap (for CS4234) – Week 09

Graph Matching

- Weighted MCBM: (Min/Max) Cost (Max) Flow, Hungarian (Kuhn-Munkres algorithm) **(high level tour)**
- Unweighted MCM: Edmonds' Matching **(high level tour)**
- Weighted MCM: DP with Bitmask **(small graph only, review…)**
  - This DP with Bitmask solution will also solve other variants, but only if they are posed on small ($V \leq 20$) graphs…
  - I am still unable to make it work for Christofides's 1.5-Approximation algorithm for M-R/NR-TSP as of year 2020
    - PS: Now 1.5-e, see https://www.quantamagazine.org/computer-scientists-break-traveling-salesperson-record-20201008/

Types of Graph Matching



Solutions (High Level Tour Only):

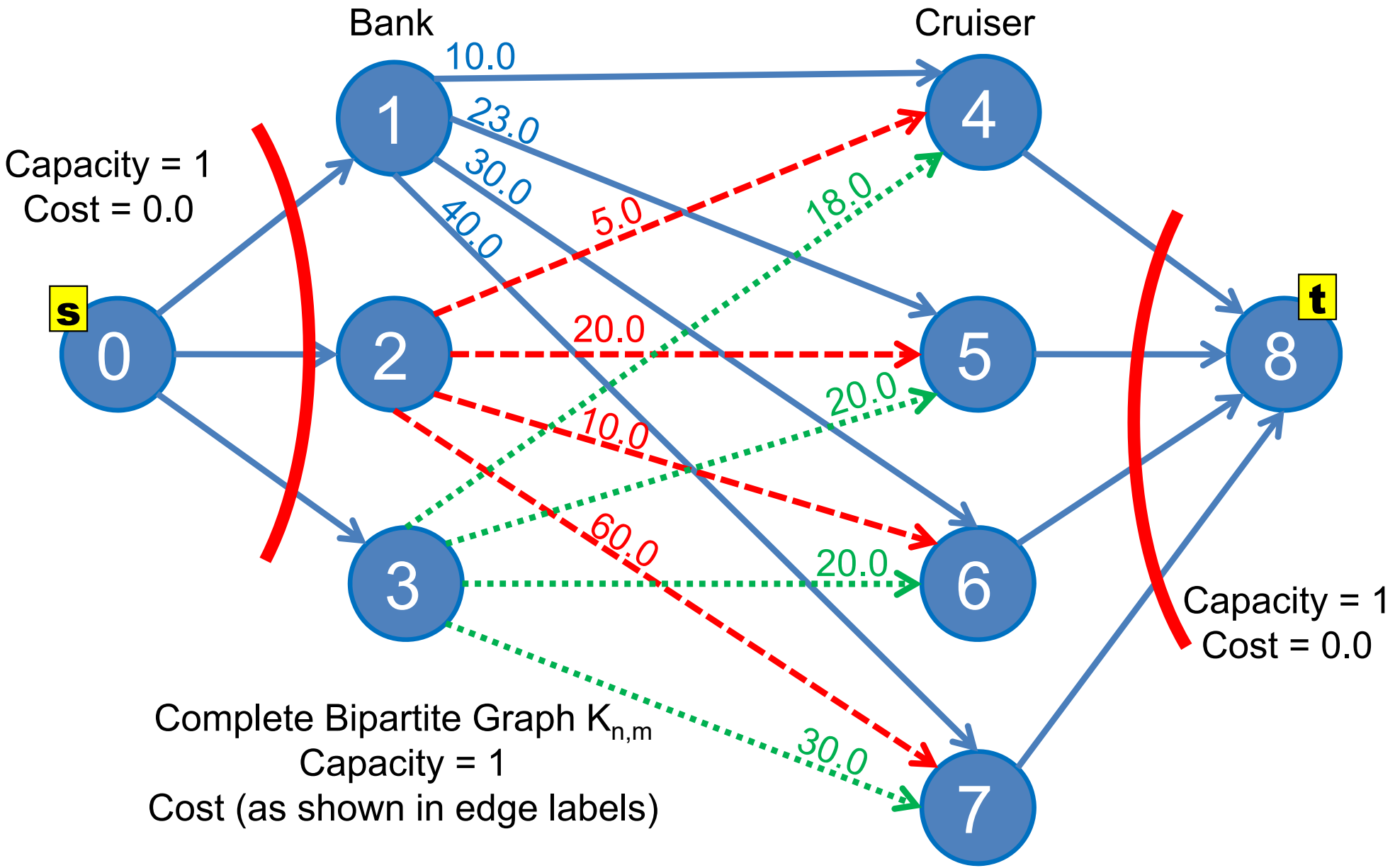
Min/Max Cost (Max) Flow, CP4 Book 2 Section 9.25

Kuhn-Munkres (Hungarian), CP4 Book 2 Section 9.27

WEIGHTED MCBM

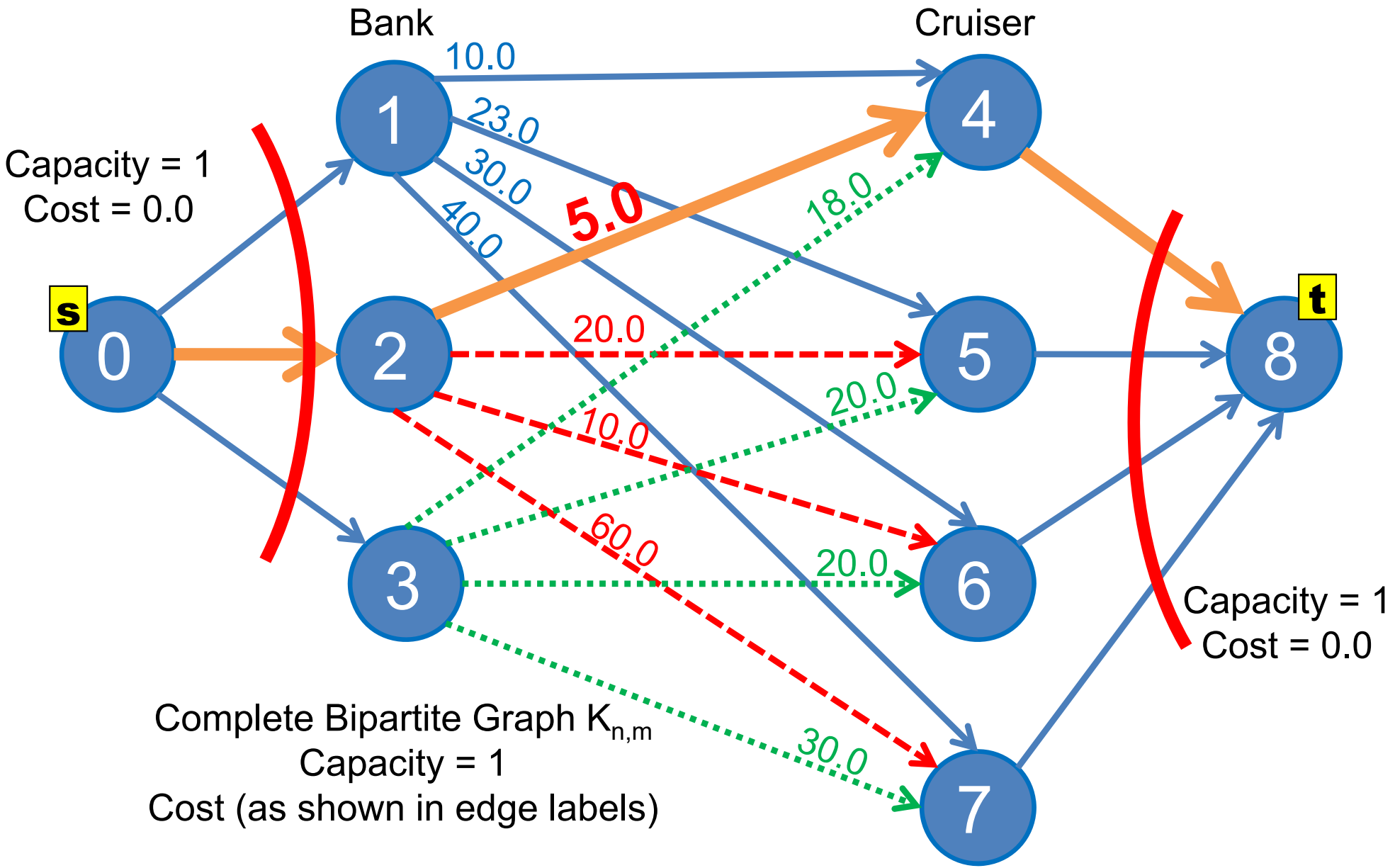
UVa 10746 (1)

Find Max Flow (=3 here)
of Minimum Cost (still in **P**)
That is, Min Cost Bipartite Matching…



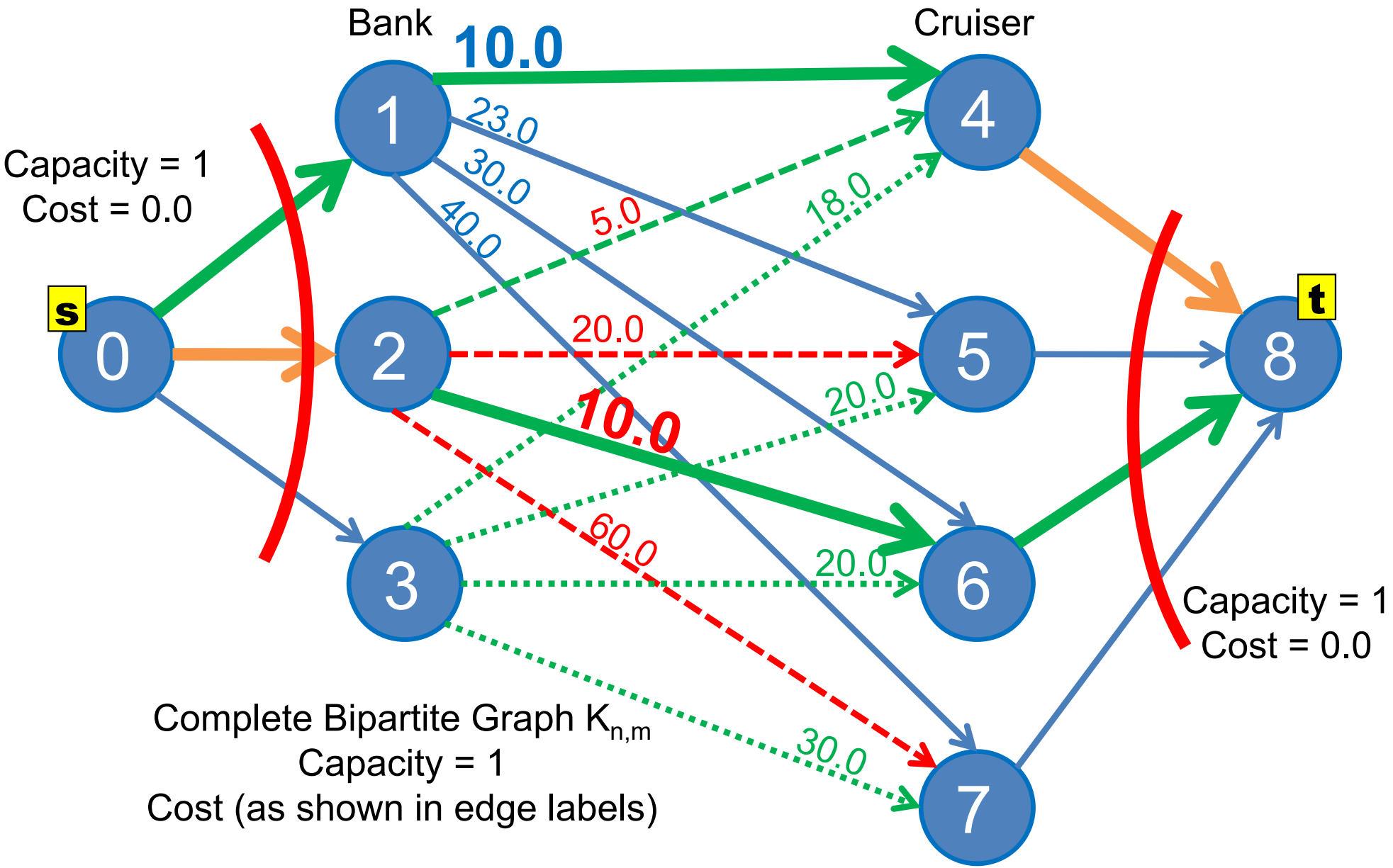
Min Cost so far =
0 + 5.0 + 0 = 5.0

UVa 10746 (2)



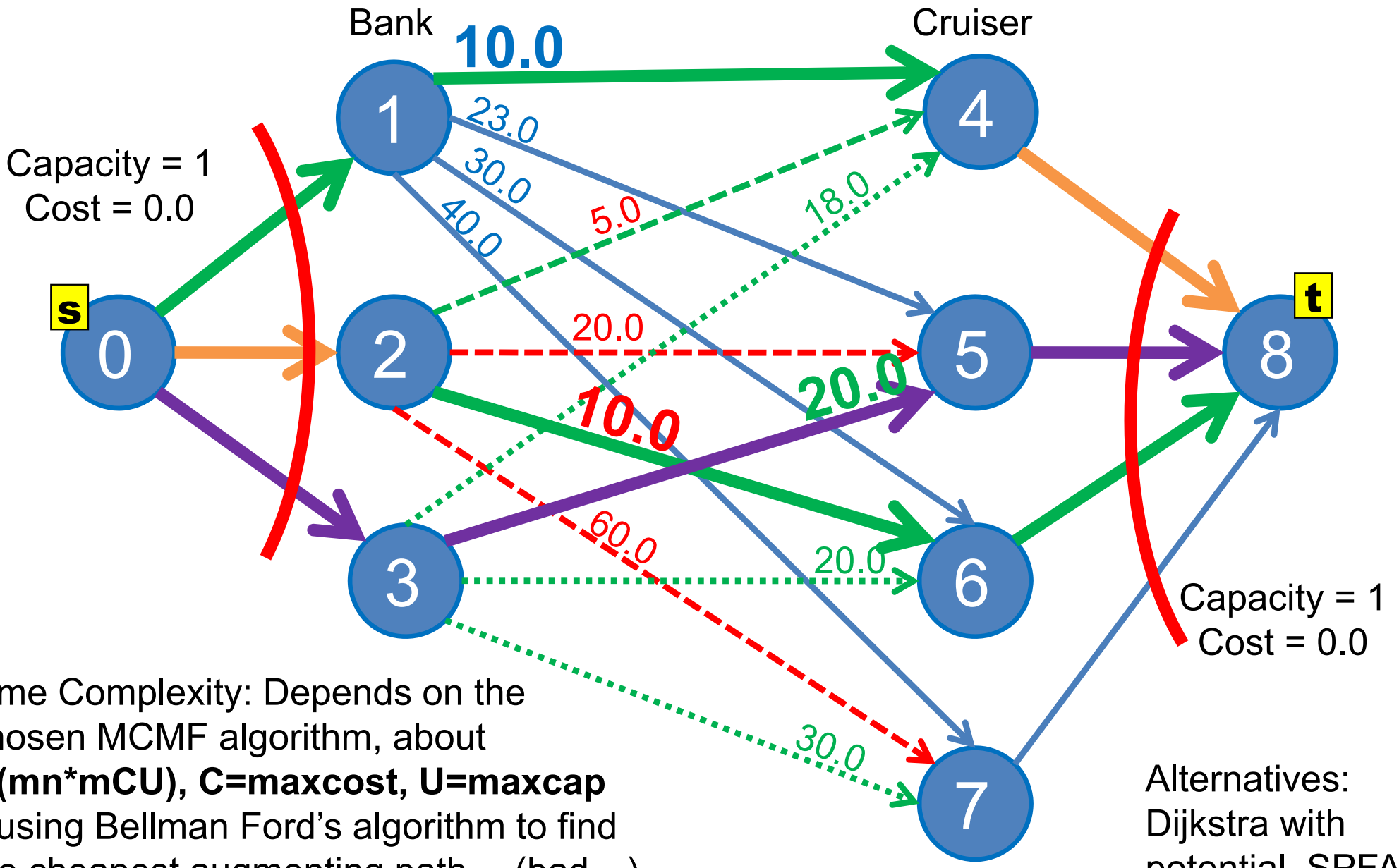
UVa 10746 (3)

Min Cost so far =
0 + **5.0** + 0 +
0 + 10.0 **- 5.0** + 10.0 + 0 = 20.0



UVa 10746 (4)

Min Cost so far =
0 + **5.0** + 0 +
0 + 10.0 **- 5.0** + 10.0 + 0 +
0 + 20.0 + 0 = 40.0



Time Complexity: Depends on the chosen MCMF algorithm, about **O(mn*mCU), C=maxcost, U=maxcap** if using Bellman Ford's algorithm to find the cheapest augmenting path… (bad…)

Alternatives: Dijkstra with potential, SPFA

Kuhn-Munkres (Hungarian) Algorithm

Harold Kuhn (1955) and James Munkres (1957) name their (joint) algorithm based on the work of two other **Hungarian** mathematicians (Denes Konig + Jano Egervary)

- There is a graph version and matrix version (we discuss graph version)

Initial implementation $O(n^4)$; today's best version $O(n^3)$

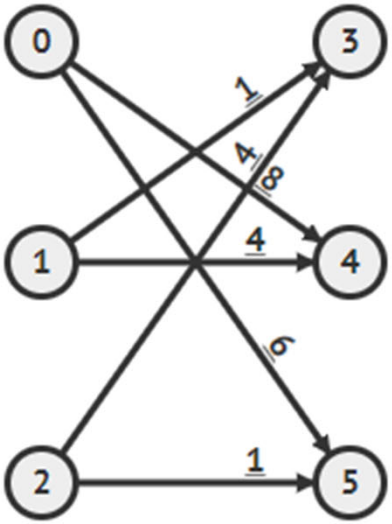
Default version: For Max Weighted Perfect Bipartite Matching

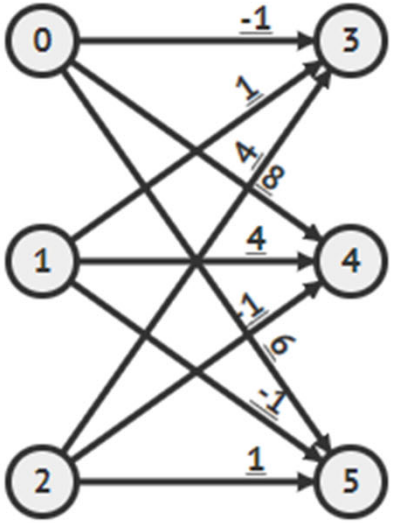
- But can easily be modified to support Min Weighted (negate edge weights) and for Bipartite Graph where Perfect Matching is impossible (add dummy vertices/edges with irrelevant weights)

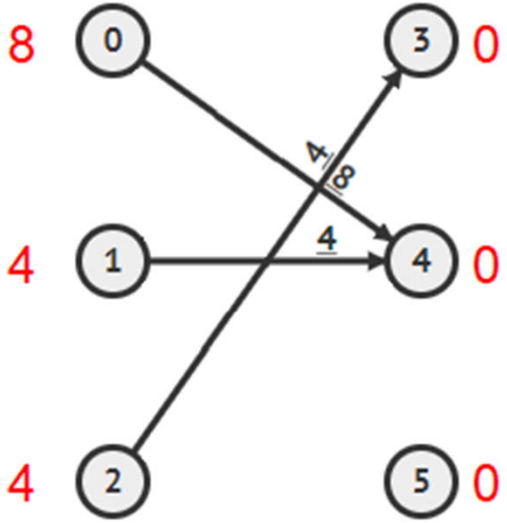
An explanation using CP4 Book 2 drawings

L: Initial Graph
M: Complete Weighted Bipartite Graph
R: Equality Subgraph

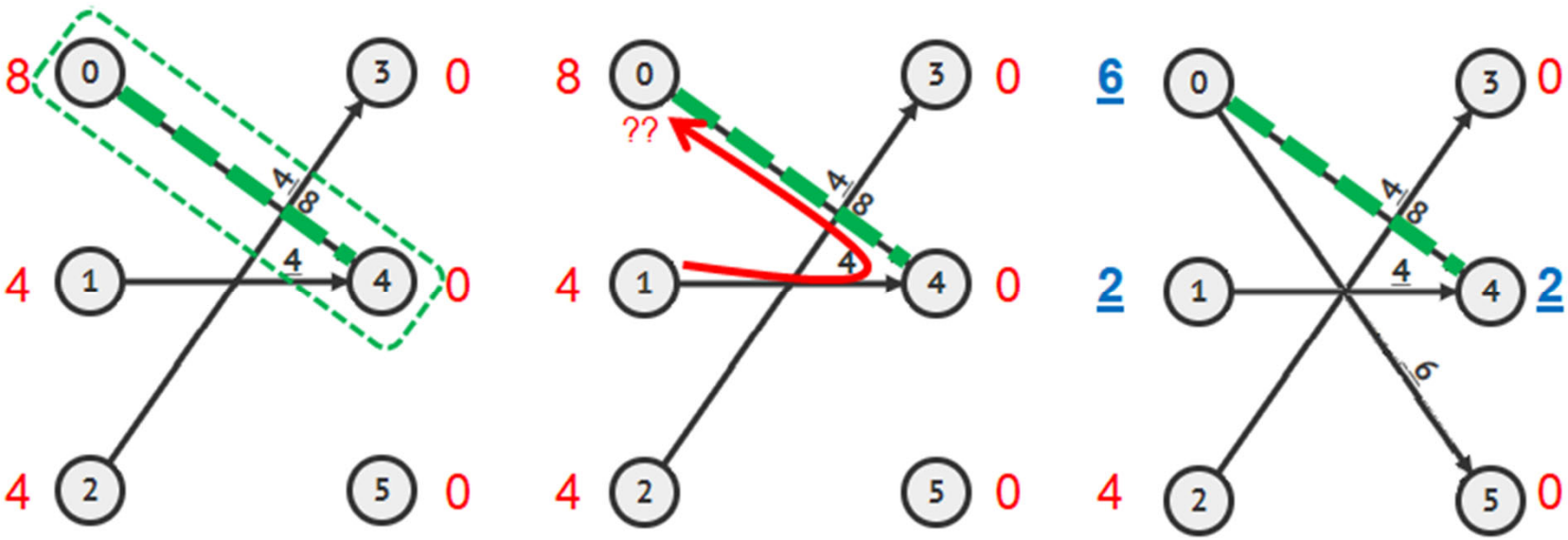
Review the recording for the explanation



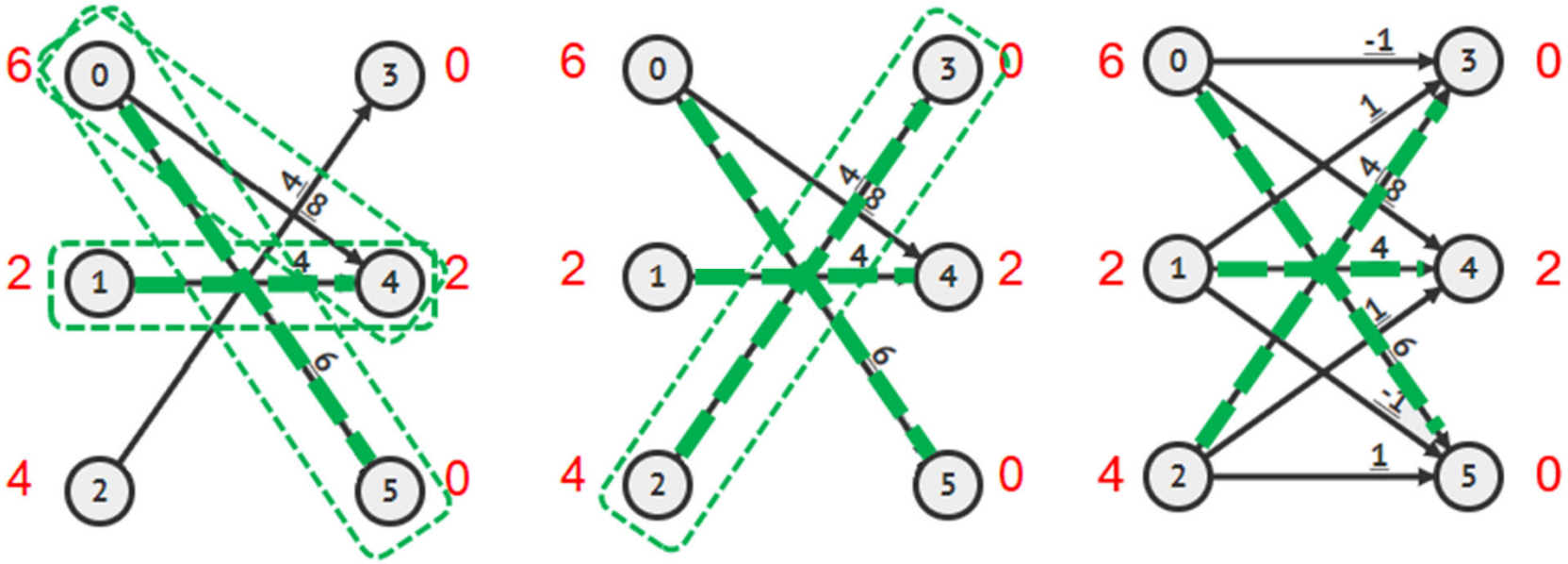




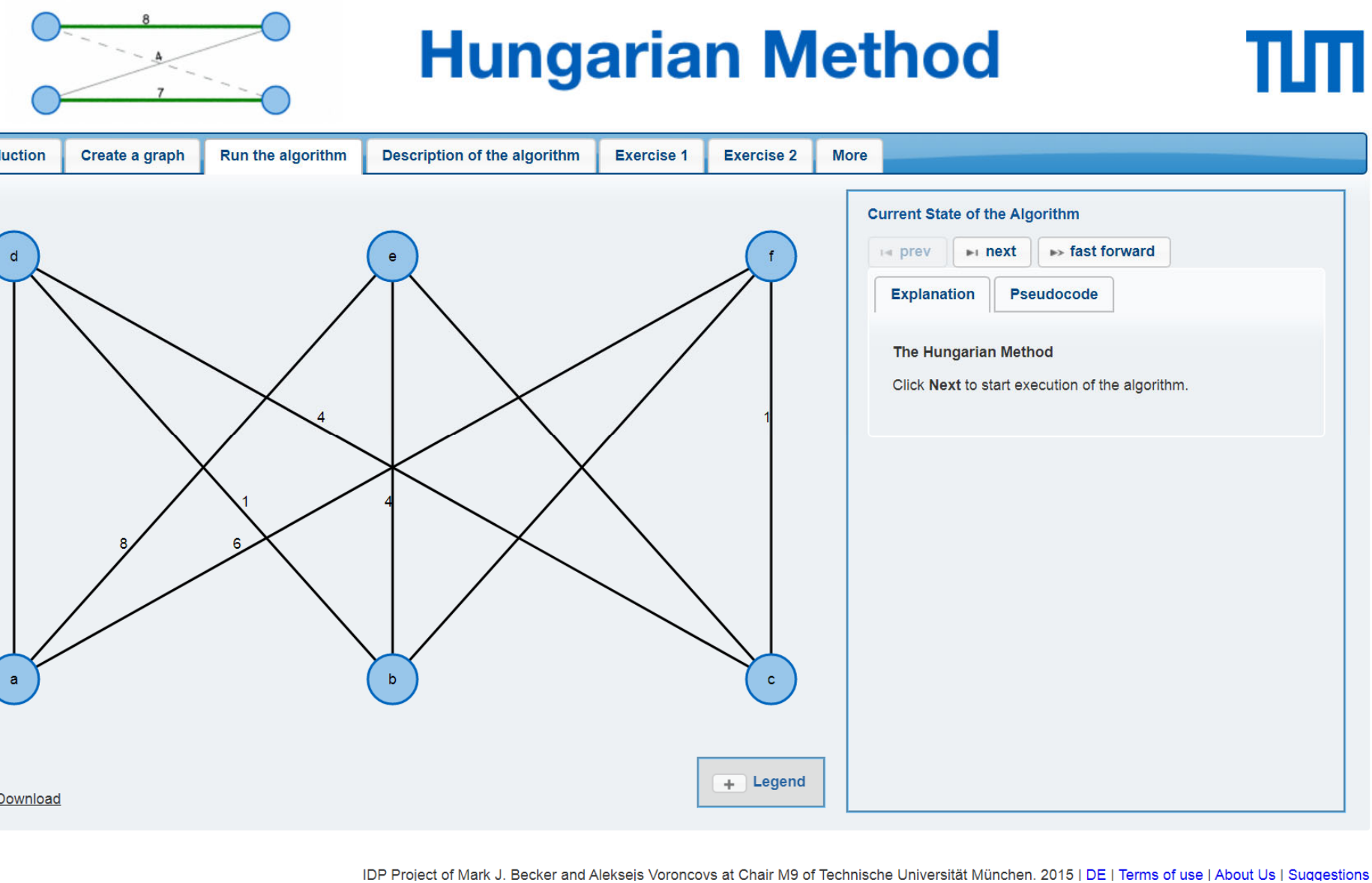
L: 1st Augmenting Path
M: Stuck
R: Relabel the Equality Subgraph



L: 2nd Augmenting Path
M: 3rd Augmenting Path
R: Max Weighted Perfect Matching



Alternatively, see



This tool https://www-m9.ma.tum.de/graph-algorithms/matchings-hungarian-method/index_en.html is on maximization problem on a **complete bipartite** graph (so that Perfect Matching exists)
The layout is top row = right set and bottom row = left set

Kuhn-Munkres (Hungarian) Algorithm

A good Hungarian algorithm implementation runs in O($\mathbf{V^3}$), thus it is a much better algorithm for **Weighted MCBM** problem compared to MCMF

- You are allowed to quote this info verbatim in exam
  - Focus on the modeling of the *complete* weighted bipartite graph
  - And on whether it is a maximization or a minimization problem
- References:
  - https://github.com/jaehyunp/stanfordacm/blob/master/code/MinCostMatching.cc
  - http://e-maxx.ru/algo/assignment_hungary
    - https://translate.google.com/translate?hl=ru&sl=ru&tl=en&u=http%3A%2F%2Fe-maxx.ru%2Falgo%2Fassignment_hungary
  - https://brilliant.org/wiki/hungarian-matching/

Solution:

Edmonds' Matching Algorithm (Overview only) CP4 Book 2, Section 9.28

DP with Bitmask (Small Graph, discussed later)

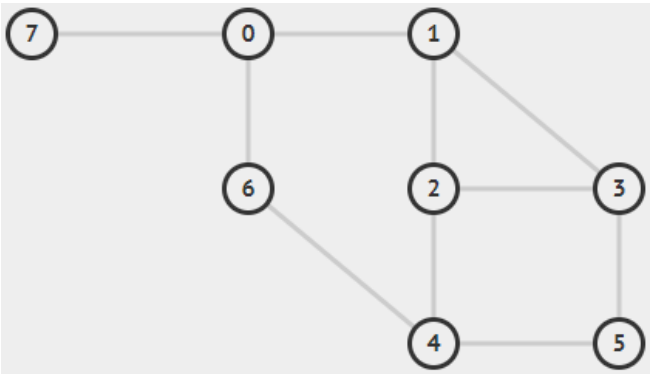
UNWEIGHTED MCM

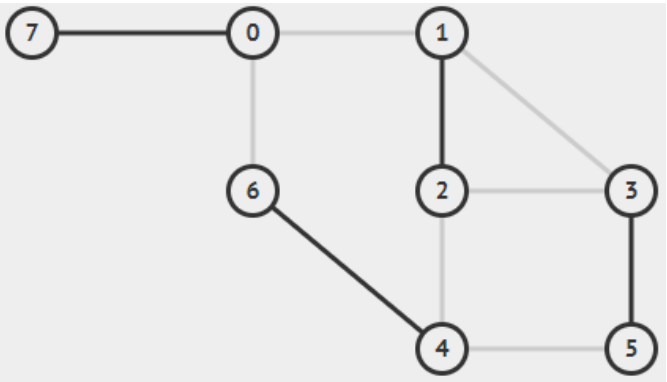
https://visualgo.net/en/matching, select “Unweighted General” tab

Non-Bipartite Graph and Blossom

A graph is not bipartite if it has at least one odd-length cycle

What is the MCM of this non-bipartite graph?





It is harder to find augmenting path (Berge’s Lemma) in such graph due to alternating cycles called **blossoms**

https://visualgo.net/en/matching, select “Unweighted General” tab

Blossom Shrinking/Expansion

Shrinking these blossoms (recursively)
will make this problem “easy” again

(Switch to draft visualization @ VisuAlgo for live explanation)

Review the recording for the explanation

When To Use Edmonds' Matching?

This algorithm is a *hard* to implement…

$O(\mathbf{V}^3)$ library code is preferred

- Only used for unweighted MCM with $\mathbf{V} \in [22..200^*$; complex code$]$
  - If $\mathbf{V} \leq [19..21]$, see this
- Randomized greedy pre-processing is ALSO APPLICABLE here!!
  - Again, you can quote this verbatim for final assessment
  - Focus on the unweighted non-bipartite graph modeling

For code, just use external source

- http://codeforces.com/blog/entry/49402 (C++)
- https://sites.google.com/site/indy256/algo/edmonds_matching (Java)

Solution(s):

DP with Bitmask (only for small graph)

Modified? Edmonds' Matching (future work…)

WEIGHTED MCM

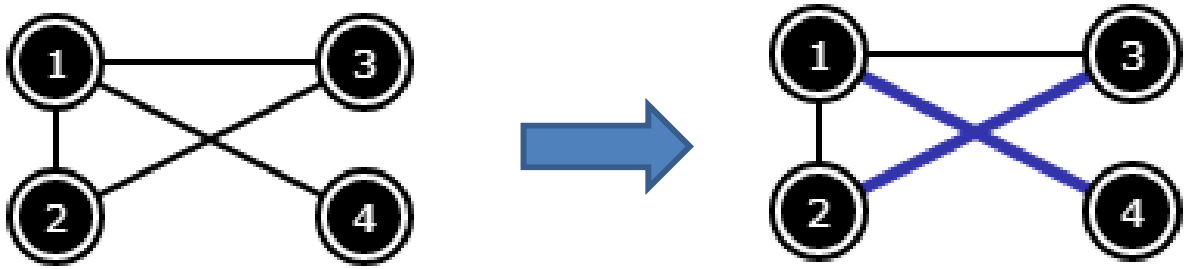
No? Choice… (only for **V** ≤ [19..21])

```
ii wMCM(int mask) {
  if (mask == (1<<N)-1) return ii(0, 0);
  if (memo[mask] != ii(-1, -1)) return memo[mask];
  int p1, p2;
  for (p1 = 0; p1 < N; p1++) if (!(mask & (1<<p1))) break;
  ii ans = wMCM(mask | (1<<p1)); // p1 unmatched
  for (p2 = p1+1; p2 < N; p2++)
    if (!(mask & (1<<p2)) && cost[p1][p2]) {
      ii nxt = wMCM(mask | (1<<p1) | (1<<p2)); // match p1-p2
      nxt.first += 2; nxt.second += cost[p1][p2];
      if ((nxt.first > ans.first) || // equal # matching
          ((nxt.first == ans.first) &&
           (nxt.second < ans.second))) // or smaller cost
        ans = nxt;
    }
  return memo[mask] = ans;
}
```

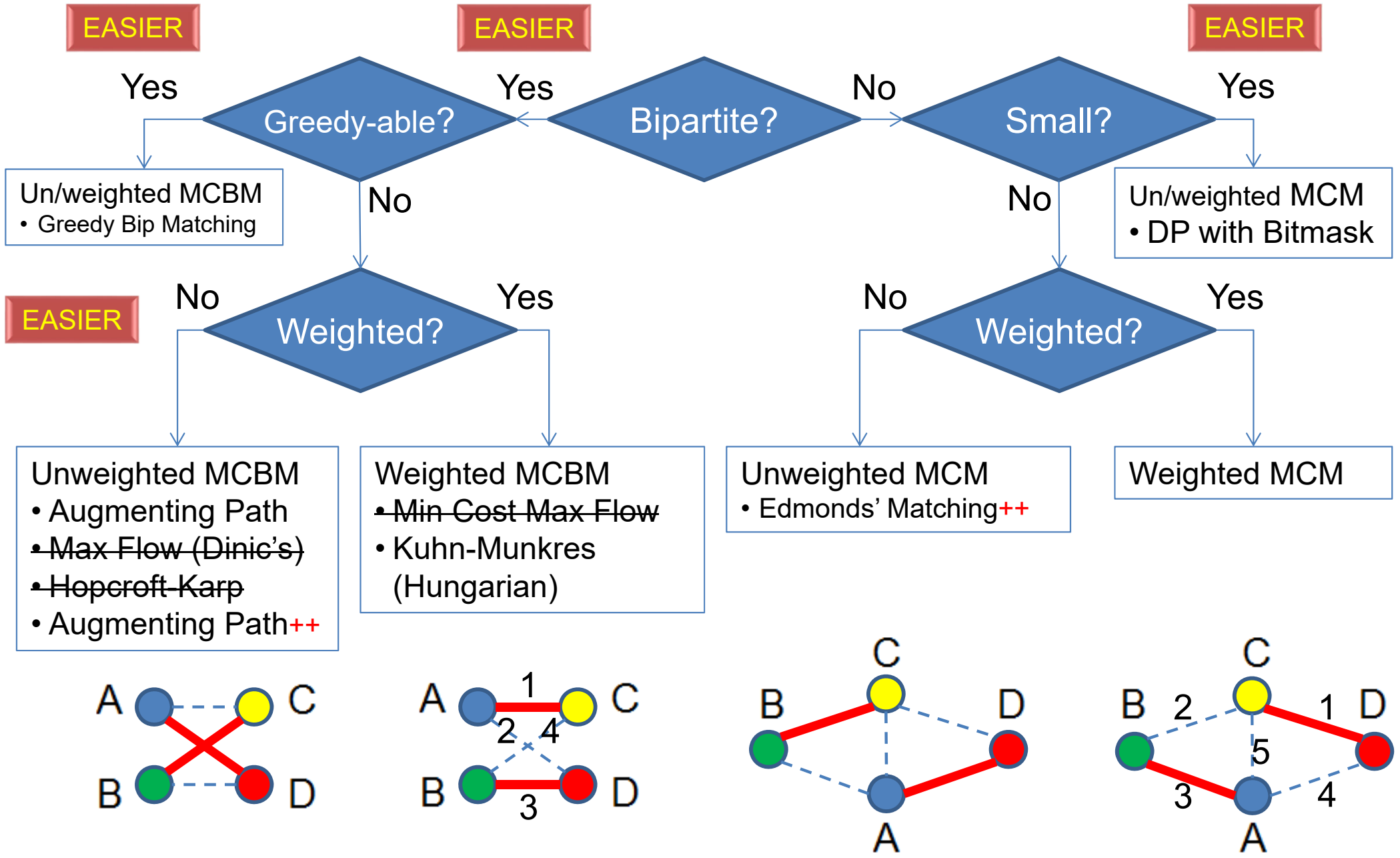
https://visualgo.net/en/recursion, select the “Matching” example…

Also for Small Unweighted MCM Too

Just set all weights = 1 in the previous code



Types of Graph Matching



References

- Mostly CP4, Book 1 Section 4.6, then Book 2 Section 8.5, and a few sections in Chapter 9 (9.25-9.29)
- TopCoder PrimePairs, RookAttack solution
- http://www.comp.nus.edu.sg/~cs6234/2009/Lectures/Match-sl-PC.pdf (Prof LeongHW's/P Karras slides)
- There are much bigger topics outside these two lectures and two tutorials and these two lecture notes will keep be improved over the years…