# Discussion 10-Supplemental

CSCI 570

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

DISCUSSION 10-SUPPLEMENTAL

# Outline

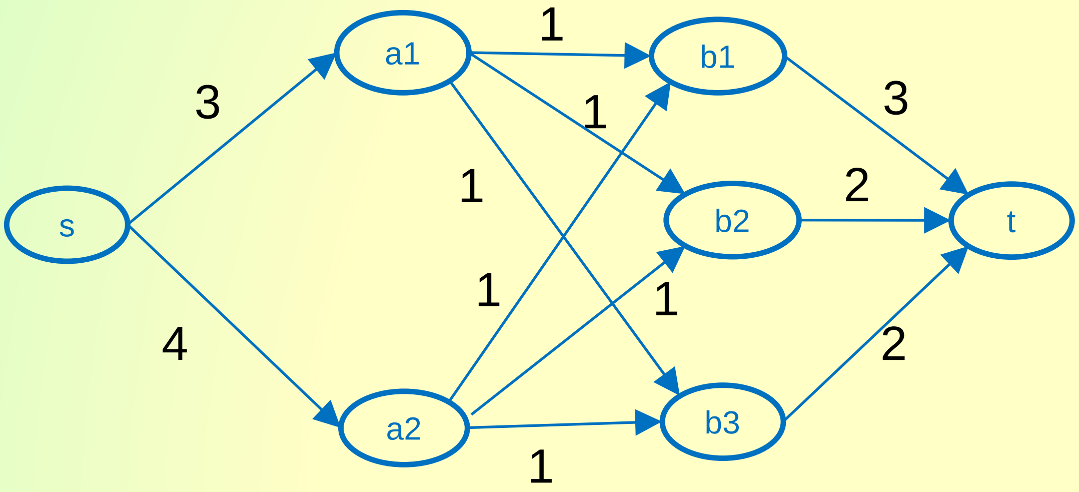
- Problems with Solutions

# Problem 3

- At a dinner party, there are *n* families $\{a_1, a_2, \ldots a_n\}$ and m tables $\{b_1, b_2, \ldots b_m\}$. The *i*$^{th}$ family $a_i$ has $g_i$ members and the j$^{th}$ table $b_j$ has $h_j$ seats. Everyone is interested in making new friends and the dinner party planner wants to seat people such that no two members of the same family are seated in the same table. Design an algorithm that decides if there exists a seating assignment such that no two members of the same family are seated at the same table.

# Solution 3

- Create a bipartite graph with one vertex for each family on one side and one for each table on the other side. Add all possible edges with capacity=1, which represents that one member of that family can sit at that table.
- Create a source s and connect s to each family vertex with capacity equal to the number of family members.
- Create a sink t and connect each table vertex to t with capacity equal to the number of seats at the table.
- If a max flow of value equal to the total number of people across all the families exists, then this is possible.


s
a1
a2
b1
b2
b3
t
3
4
1
1
1
1
1
1
3
2
2