Announcements

- HW 6 (the last homework!) Due on Sunday
- Please Remember to fill out your CAPEs
- If you cannot take the final at the usual time please let me know by Wednesday

Last Time

- Network – Directed graph G with source (s) and sink (t).
- Flow – subgraph so that $d_{in}(v)=d_{out}(v)$ for all v not s, t.
  - Size(F) = $d_{out}(s)-d_{in}(s) = d_{in}(t) – d_{out}(t)$
- Augmenting Paths
  - Path from s to t uses unused edges in forward direction or used edges in backward direction can be used to increase flow.
- Cut – partition of the vertices $s \in S$, $t \in T$
  - Side = #{edges S to T}
- Size(F) = #{edges of F from S to T} – #{edges of F from T to S}

Today

- Maxflow-Mincut
- Applications
- General Perfect Matchings

Maxflow-Mincut

Theorem (V. 8.3.2): For any network G the size of a maximum flow in G is the same as the size of a minimum cut.

Idea

- A cut is a bottleneck. No flow can be more than any cut.
- Suppose that as much traffic is leaving the city as possible (maxflow). You try to escape in your car and find yourself blocked by traffic. The roads that are full describe a matching cut.

Maxflow ≤ Mincut

Let F be a flow and C be a cut.

By Lemma:

$$\text{Size}(F) = \#\{\text{edges in F from S to T}\} - \#\{\text{edges in F from T to S}\} \leq \text{Size}(C)$$

Any flow is smaller than any cut, so the maximum flow size is at most than the minimum cut size.

Maxflow ≥ Mincut

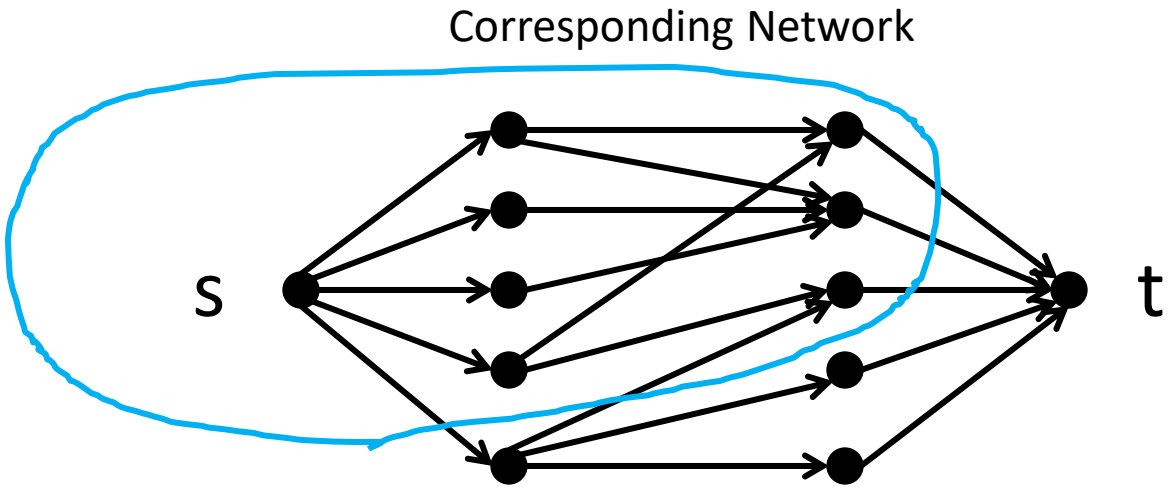
- Let F be a maximum flow.
- No augmenting paths.
- Let S be the set of vertices v you can reach from s using unused forward edges or used backwards edges.
- F uses all edges out of S, no edges into S.
- Lemma says: Size(F) = Number of edges out of S = Size(C).
- Maxflow ≥ Mincut.

Capacities

Everything still works if we allow multiple edges. This basically corresponds to some routes having a higher capacity than others (if there are two edges from u to v, you can run two units of flow there). You can also generalize further by giving each edge a real numbered "capacity" for how much flow it can handle.

Maxflow-Mincut and Konig's Theorem

Konig's Theorem can be thought of as a special case of Maxflow-Mincut.



Matchings…
Correspond to flows.

Vertex covers…
Correspond to cuts.

Menger's Theorem

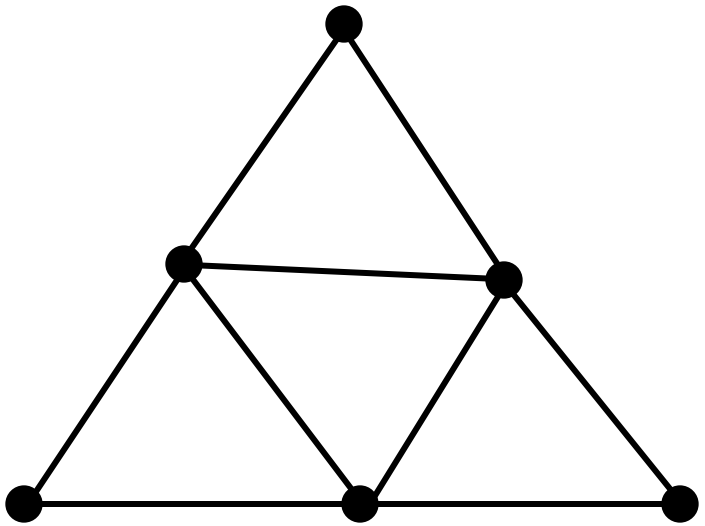
- Maxflow-Mincut *is* the edge version of Menger's Theorem. It says that the minimum number of edges you need to remove to disconnect s from t (the smallest cut size), is the maximum number of edge-disjoint paths (the maximum flow size).
- You can also use Maxflow-Mincut to prove the vertex Menger's Theorem with extra work.

Perfect Matchings of General Graphs

Hall's Theorem tells us when a bipartite graph G has a matching that uses up all the vertices in a bipartite graph. Although this is a common application of matchings, it is sometimes useful to think about when such *perfect matchings* exist in more general graphs.

Perfect Matchings

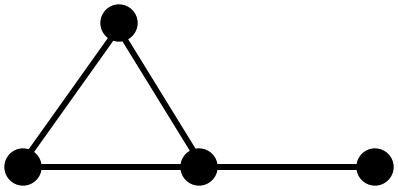
- Given graph G
- Find a set of edges that uses each vertex once.



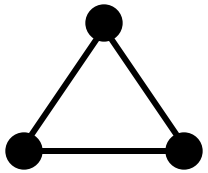
Question: Perfect Matching

Which of these graphs have a perfect matching?

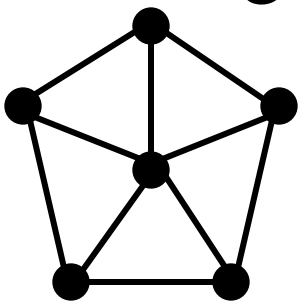
A



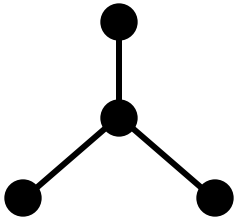
B

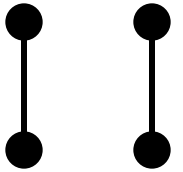


C



D





E

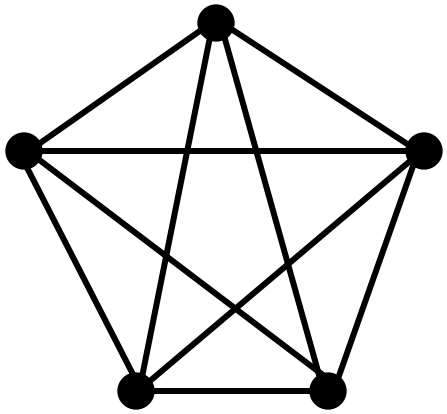
Obstacles

In order to classify which graphs have perfect matchings and which don't, we need to understand what obstacles there might be to having one.

Example I

K_5 has no perfect matching.

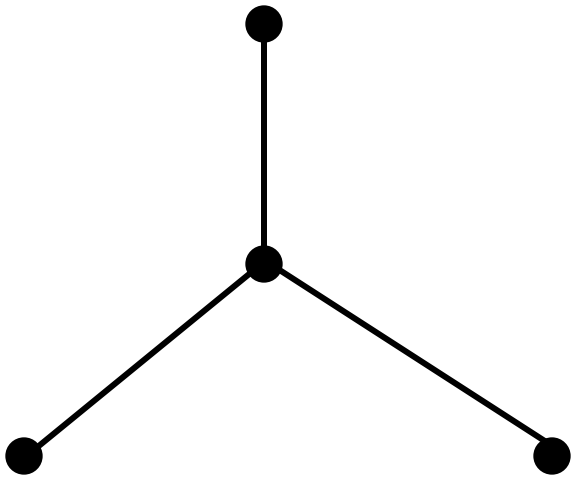
Any matching uses an *even* number of vertices.



Example II

$K_{1,3}$ has no perfect matching

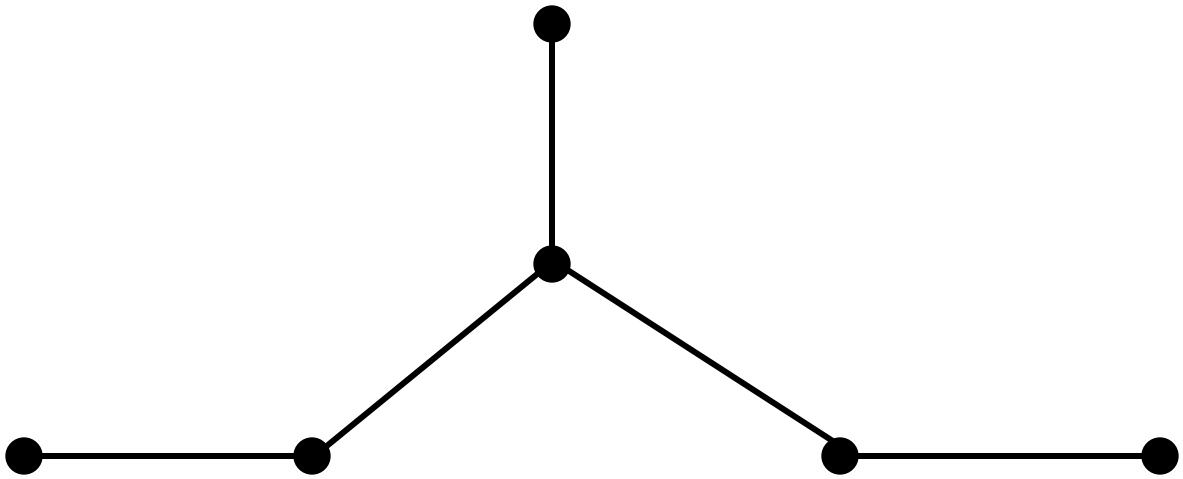
All the other vertices need to pair with the middle vertex



Example III

This tree *does* have a perfect matching.

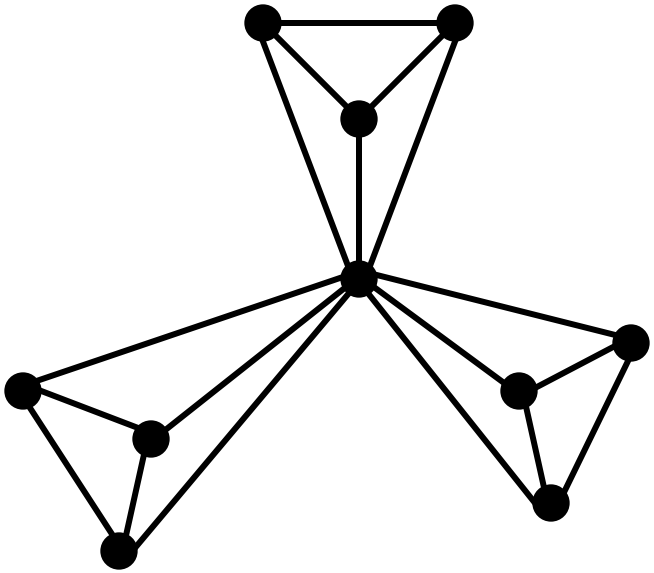
The other vertices provide relief for the middle.



Example IV

This graph does not.

The central vertex only connects to one component, the other two are odd.



Example V

Again no matching.

The two central vertices can only help out two of the four other components.