

Exercises Week 6

1 Understanding the definitions

1. Show that K_5 is not planar.
2. What is the difference between a planar graph and a plane graph?
3. Let G be the graph whose vertices are ordered triples of zeros and ones, and two vertices adjacent if they only differ in one coordinate. Show that this graph is planar.
4. How many faces are there in the graph from the previous exercise? Does this depend on the graph drawing?

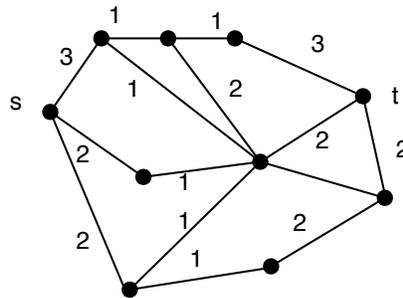
2 Exercises

1. Create an algorithm out the proof of Theorem 6.2.2 and calculate the number of steps it may take (in the worst case scenario).
2. Prove Konig's theorem using the max-flow min-cut theorem.
3. Using the max-flow min-cut theorem to show Menger's Global theorem. For the vertex version of it, construct an auxiliary directed graph and assume that Ford Fulkerson works for directed graphs.
4. In case you have not proved last week's bonus problem, do it now: Prove Hall's theorem using Menger's theorem.
5. The Edmonds-Karp algorithm for maximum flow minimum cut is the same as the Ford-Fulkerson with the exception that at every iteration the new path chosen must be of the shortest size possible (for example using the Breadth-first-search algorithm seen in class). Show that the number of steps taken by this algorithm is $O(nm^2)$ where n and m are the number of vertices and edges, respectively.

Hint: In order to prove this, consider the levels of the breadth-first-search algorithm. Let L_i are the vertices at distance i . Note that every time the algorithm takes a new shortest path then it will contain

exactly one vertex from each L_i . Analyse what happens when the graph is updated to account for the new flow. Show that at each iteration the distance from the sink to any vertex never decreases. Conclude that every time the distance from the saturated edge to the source along the augmenting path must be longer than last time it was saturated, and that the length is at most n . Don't forget to put everything together!

6. In the following network find a maximum flow from s to t and a minimum st -cut USING Ford Fulkerson's algorithm.



Bonus Problem: Find an algorithm (that runs in a reasonable amount of time) that finds, for every bipartite graph, a maximum independent set of vertices.