COSC 6111 Advanced Design and Analysis of Algorithms
Jeff Edmonds
Assignment: Network Flow

First Person:
Family Name:
Given Name:
Student #:
Email:

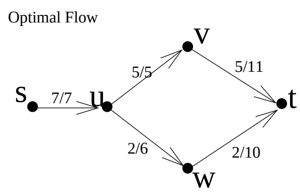
Second Person:
Family Name:
Given Name:
Student #:
Email:

Do 2 and 3.

Problem Name	If Done Old Mark	Check if to be Marked	New Mark
1 Improve Edge			
2 Hall's Theorem			
3 Running Time			

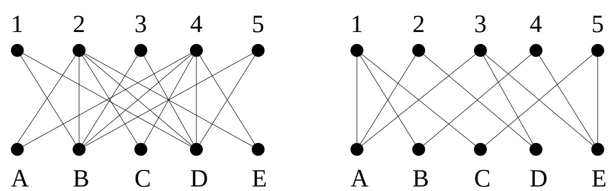
COSC 6111 Advanced Design and Analysis of Algorithms
Jeff Edmonds
Assignment: Network Flow

1. Network Flow: Suppose you are given a network flow instance, an optimal flow for it, and an edge $\langle u,v \rangle$. Though the current flow through the edge $\langle u,v \rangle$ does not exceed its capacity, the owner of this edge has bribed the authorities to decrease the flow through this edge as much as possible while maintaining the validity and the optimality of the overall flow. You are to design an algorithm for this problem. Explain your algorithm using the following instance as an example. A formal proof is not needed. What algorithms learned in class do you use to find the required structures?



2. Let $G = (L \cup R, E)$ be a bipartite graph with nodes L on the left and R on the right. A matching is a subset of the edges so that each node appears at most once. For any $A \subseteq L$, let $N(A)$ be the neighborhood set of A, namely $N(A) = \{v \in R \mid \exists u \in A \text{ such that } (u,v) \in E\}$. Prove Hall's Theorem which states that there exists a matching in which every node in L is matched if and only iff $\forall A \subseteq L,\ |A| \leq |N(A)|$.

   (a) For each of the following two bipartite graphs, either give a short witness to the fact that it has a perfect matching or to the fact that it does not. Use Hall's Theorem in your explanation as to why a graph does not have a matching. No need to mention flows or cuts.



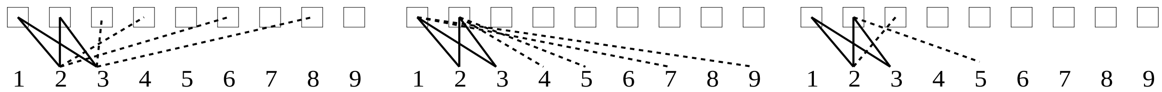
   (b) $\Rightarrow$: Suppose there exists a matching in which every node in L is matched. For $u \in L$, let $M(u) \in R$ specify one such matching. Prove that $\forall A \subseteq L,\ |A| \leq |N(A)|$.

   (c) Look at both the slides and section 19.5 of the notes. It describes a network with nodes $\{s\} \cup L \cup R \cup \{t\}$ with a directed edge from s to each node in L, the edges E from L to R in the bipartite graph directed from L to R, and a directed edge from each node in R to t. The notes gives each edge capacity 1. However, The edges $\langle u,v \rangle$ across the bipartite graph could just as well be given capacity ∞.

   Consider some cut (U,V) in this network. Note U contains s, some nodes of L, and some nodes of R, while V contains the remaining nodes of L, the remaining nodes of R, and t. Assume that $\forall A \subseteq L,\ |A| \leq |N(A)|$. Prove that the capacity of this cut, i.e. $cap(U,V) = \sum_{u \in U} \sum_{v \in V} c_{\langle u,v \rangle}$, is at least $|L|$.

   (d) $\Leftarrow$: Assume that $\forall A \subseteq L,\ |A| \leq |N(A)|$ is true. Prove that there exists a matching in which every node in L is matched. Hint: Use everything you know about Network Flows.

   (e) Consider the game Sudoku and the following three figures. The nine squares are those in some row, column, or 3×3 square. The edges indicate which numbers are allowed to go into which squares. Only the edges involving the first two squares and/or the numbers 2 and 3 are given. The solid lines are more significant to this question than the dotted ones. For each of these figures, explain what you can say about which numbers can be mapped to which holes. For example, might 2 go in the forth square?



3. Suppose you have an algorithm which maintains a vector of positive integers whose values at time t are $A_t = \left\langle a_{\langle t,1\rangle}, a_{\langle t,2\rangle}, \ldots, a_{\langle t,n\rangle} \right\rangle$. Initially, each $a_{\langle 0,i\rangle}$ is an ℓ bit integer. Each iteration some of the entries may decrease while others increase. To bound the running time, we define the measure S_t of the vector to be $S_t = S(A_t) = \sum_i \left(a_{\langle t,i\rangle}\right)^r$ for some parameter r. What is guaranteed is that the current value decreases by at least $\Delta_t = \left(a_{\langle t,i_{max}\rangle}\right)^r$ where $a_{\langle t,i_{max}\rangle}$ is the largest entry in A_t. The algorithm stops when each entry is zero. Give the bound on the number of iterations of this algorithm as a function of the parameters n, ℓ, and r.