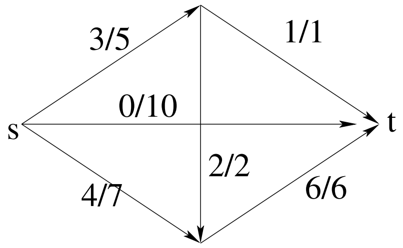
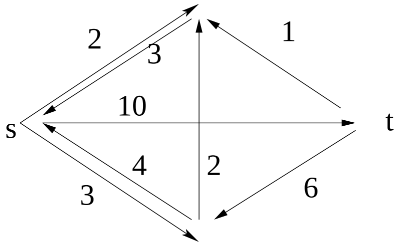
CMPT 307 Solutions to Quiz 4 Dec 1, 2011

[3] 1. (a) For a given flow network G with flow f, draw the residual network G_f and show the residual capacities of all edges in G_f. (In the graph below, the edge label a/b means there is a flow of value a along the edge, and the edge capacity is b.)



Solution:



[3] (b) What is the flow value for the network shown above? Is this flow maximum? If not, what would be the maximum flow?

Solution: The flow value is 7. It is not maximum. The maximum would be 17 (with extra 10 flow on the edge from s to t).

[3] (c) What is the minimum-capacity st-cut for the network shown above? Show the cut. What is the capacity of your cut?

Solution: The st-cut would contain s and two middle points on one side, and just t on the other side. It has capacity $1 + 10 + 6 = 17$.

[3] (d) Argue why your st-cut from the previous item is of minimum capacity.

Solution: It is the min cut since there is flow of value 17 equal to the capacity of that cut, and the claim follows by the MaxFlow=MinCut theorem. (We know that every flow can have value at most min cut capacity; since we have a cut of capacity 17 and a flow of value 17, we conclude that this cut has min capacity.)

2. **Committee assignments**:

At a certain university there are n professors and m committees that need to be formed. Each professor i has a list of committees L_i where professor i may serve. Also, each professor may serve on at most one committee. Finally, each committee may have at most 5 professors serving on it. The president of the university wants to assign every professor to do some committee work.

Give an efficient algorithm that would decide if it is possible to assign all n professors to some committee work, subject to all the constraints specified above.

[5] (a) Explain how to build a flow network for the committee assignment problem.

Solution: For each professor i, there is a node p_i; also, for each committee j there is a node c_j. We connect s to each node p_i, $1 \leq i \leq n$, via an edge of capacity 1. We connect each node p_i to the committee nodes that are in L_i, via edges of capacity 1. Finally, we connect each committee node c_j to t via an edge of capacity 5.

[5] (b) Explain how to check if an assignment of all n professors to some committees is possible, using your flow network from the previous item.

Solution: The max flow has value n iff there is a committee assignment where each professors gets to do some committee work.

[5] (c) Finally, explain how to get a committee assignment that would maximize the number of professors assigned to do some committee work.

Solution: Run the MaxFlow algorithm. Get a max flow of some integer value k, where $0 \leq k \leq n$. Take k professors whose nodes get incoming flow. Assign each such professor to the unique committee node that receives inflow of 1 from the professor node.

3. The Ford-Fulkerson algorithm finds an augmenting path in a current residual network, updates the flow along the found path, and repeats, until no augmenting paths exist.

[3] (a) How many augmentation iterations are possible during the run of the Ford-Fulkerson algorithm in the worst case?

Solution: $C = \sum_{e \text{ out of } s} c(e)$, i.e., the sum of the capacities of all edges leaving from s.

[3] (b) How is the Edmonds-Karp algorithm different from the standard Ford-Fulkerson algorithm?

Solution: Whenever an augmenting path needs to be chosen, we choose a shortest augmenting path (with fewest edges).

[3] (c) How many augmentation iterations are possible during the run of the Edmonds-Karp algorithm?

Solution: $O(mn)$, where m is the number of edges in the original graph, and n is the number of nodes.

4. Complete the following definitions:

[3] (a) A decision problem A is in NP if

Solution: there is a polynomial p and a polytime algorithm V such that, for every input x, $x \in A$ iff $\exists y$, $|y| \leq p(|x|)$ & $V(x, y)$ accepts.

[3] (b) A decision problem A is polytime reducible to B ($A \leq_p B$) if

Solution: there is a polytime mapping $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ such that, for every input x, $x \in A$ iff $f(x) \in B$.

[3] (c) A decision problem A is NP-complete if

Solution: *(i)* $A \in NP$, and *(ii)* for every $B \in NP$, we have $B \leq_p A$.

5. **NP-completeness** Consider the following problem SQRT-IS: Given a graph $G = (V, E)$ on n vertices, decide if G has an independent set of size at least $\sqrt{n}$. Prove that SQRT-IS is NP-complete. For the reduction, you may use the problem IS (given a graph G and an integer k, decide if G has an independent set of size at least k).

[5] (a) Show that SQRT-IS is in NP.

Solution: Given $G = (V, E)$ is in SQRT-IS iff there is a subset $S \subseteq V$ of size $|S| \geq \sqrt{n}$ such that no pair of nodes from S are connected by an edge in G. Thus, we can take as y (in the definition of NP) the description of a set S. The size of y is at most n (by representing S by a binary string of length n). Checking that S is of the right size and doesn't contain an edge can be done in time polynomial in n (in the straightforward way). Hence, SQRT-IS is in NP.

[7] (b) Give a reduction IS $\leq_p$ SQRT-IS. Just give an algorithm for mapping IS instances to SQRT-IS instances. Don't argue the correctness of your reduction yet.

Solution: Given an instance (G, k) of IS, we create an instance G' of SQRT-IS as follows. The new graph G' will contain a copy of the original graph G (on n nodes), plus $n^2 - n$ new nodes. Among the new nodes, we take a subset T of $n - k$ nodes, and the subset Q of the remaining $n^2 - n + k$ new nodes. We add edges between every node in Q and every node in the graph G'. (Thus, in particular, Q forms a clique in G'.) Note that the new graph G' has n^2 nodes.

[8] (c) Prove that your reduction in the previous item is correct.

Solution: We need to argue that G has independent set of size at least k iff G' has an independent set of size at least n. We need to show both directions.

For one direction, suppose A is a set of k nodes in G that is an independent set. Take the copy of that set A inside G' (which contains a copy of G), and add to it the set T of “new” nodes of G'. The resulting set has size n, and is an independent set by construction of G'.

Conversely, suppose G' has an independent set of size n. This independent set may contain at most $n - k$ new nodes of G'; any set of more than $n - k$ new nodes will contain an edge. Thus, this independent set in G' has at least $n - (n - k) = k$ old nodes that form an independent set of size k. These old nodes correspond to an independent set of size k in G. Thus, G has an independent set of size k.