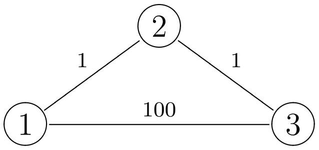
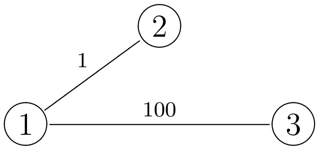
Network Optimization (Fall 2008) — Brief Solutions to Practice Final Exam

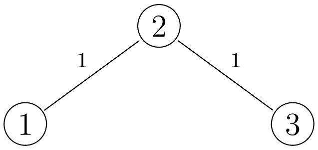
1. If G^0 has a cycle, then we can obtain an alternative optimal solution to the original problem by starting with $\boldsymbol{x}^*$ and augmenting one (or more) units of flow along the same cycle in $G(\boldsymbol{x}^*)$.

   Conversely, let $\boldsymbol{x'}$ be an alternative optimal solution. Then the difference in the flows $\boldsymbol{x}^* - \boldsymbol{x'}$ can be decomposed into at most m cycles in $G(\boldsymbol{x}^*)$. Since both flows are optimal, we must have $\boldsymbol{c}(\boldsymbol{x}^* - \boldsymbol{x'}) = 0$. By the optimality of $\boldsymbol{x}^*$, all arcs in $G(\boldsymbol{x}^*)$ must have non-negative reduced costs, and hence the reduced cost of every arc in these cycles must be zero. As such, these cycles are also present in G^0.

2. The Divide and Conquer algorithm will not work. For a counterexample, consider the following graph. The algorithm could end up choosing $N_1 = \{1, 3\}$ and $N_2 = \{2\}$, in which case the resulting spanning tree will not be an MST, as it will pick the arc $(1, 3)$. The MST will not contain this arc.







3. (a) Any s-t path of length k needs *at least* k directed s-t cuts to be covered. Hence we immediately get the corresponding inequality – the maximum length of an s-t path cannot be more than the minimum number of directed s-t cuts that cover all arcs that are on at least one s-t path.

   To get equality, let $d(i)$ represent the length of the longest path (in terms of number of arcs) from s to i for all $i \in N$. Let $k = d(t)$, the maximum length of an s-t path. For $j = 1, \ldots, k$, consider $N_j = \{i \in N \mid d(i) < j\}$, and the corresponding directed cuts $\mathcal{A}(N_j)$. These form k directed cuts that cover all the arcs that are on at least one s-t path.

   (b) The weighted version is understood as an application of the above case, where $c_{ij} = 1 \, \forall \, (i, j) \in A$. In the general case, when $c_{ij} > 1$ (and integer), we replace the arc (i, j) by a directed path having c_{ij} arcs, each with unit weight (by adding $c_{ij} - 1$ extra nodes, and the arcs in between them). If $c_{ij} = 0$, then we can remove that arc (such an arc will not contribute towards the length of a path directed from s). We then apply the result from part (a) to this modified graph. Thus, we will get at least $c_{ij} \geq 1$ directed cuts corresponding to the arc (i, j) in the original network (with weight c_{ij}). All these cuts can be grouped together, and thought of as "covering" the original arc c_{ij} times. Under this analysis, we get the following result:

   The maximum length of an s-t path is equal to the minimum number of directed s-t cuts covering each arc that is on at least one s-t path, *at least* c_{ij} *times*.

4. Notice that in a k-regular graph, we must have $|N_1| = |N_2| = n$ (say), since the sum of the degrees of all nodes in N_1 must equal the sum of the degrees of all nodes in N_2, and since the degree of each node is k. It is easy to prove the result in this question (every k-regular graph has a perfect matching) by just considering the correspondence between a maximum flow and a perfect matching. The point here is to use the *max-flow min-cut theorem* to prove the result. Hence, we will prove the minimality of the corresponding cut from first principles.

   Consider the graph $G' = (N', A')$ with $N' = N_1 \cup N_2 \cup \{s, t\}$ and $A' = A \cup \{(s, i) \,\forall\, i \in N_1\} \cup \{(j, t) \,\forall\, j \in N_2\}$. Set $u_{ij} = 1 \,\forall\, (i, j) \in A'$. We show that the cut $[\{s\},\, N' \setminus \{s\}]$ is a min-cut. The capacity of this cut is n. Label the nodes in N_1 as $\{1, \ldots, n\}$, and those in N_2 as $\{1', \ldots, n'\}$. Now consider the s-t cut $[S, \overline{S}]$ where $S = \{s\} \cup \{1, \ldots, p\} \cup \{1', \ldots, q'\}$. There will be at least $k|p - q|$ arcs from A (the original set of arcs) in this cut. Thus, there *could* be none of the original arcs crossing the cut if $p = q$. There are $n - p$ arcs in this cut that are of the form (s, i) for $i \in \{p + 1, \ldots, n\}$, and there are q arcs of the form (j', t) for $j' \in \{1, \ldots, q'\}$. The capacity of this cut will be at least

$$\begin{array}{ll} n + (k-1)(p-q) & \text{if } p > q, \quad \text{and at least} \\ n + (k+1)(q-p) & \text{if } p < q. \end{array}$$

   Thus, the capacity is at least m for $k \geq 1$. Notice that we cannot have an s-t cut if $p = q < n$. Thus, $[\{s\}, N' \setminus \{s\}]$ is a min-cut. Hence, by the max-flow min-cut theorem, the max-flow value is n, which corresponds to a perfect matching.

   Alternately, one could set $u_{ij} = \infty$ for all $(i, j) \in A$. In this case, there are exactly two minimum s-t cuts – $[\{s\}, N' \setminus \{s\}]$ and $[N' \setminus \{t\}, \{t\}]$, both with capacity n. Since the original arcs all have infinite capacity, they cannot be part of a min-cut. To complete the proof, one needs to argue that any flow in the original arcs A corresponds to a matching, and hence a max-flow corresponds to a perfect matching.

5. Notice that if $\sum_i s_i \neq n(n-1)/2$, then the set of scores is infeasible to start with. We can model this problem as checking the feasibility of a transportation problem. Create two sets of nodes $N_1 = \{1, 2, \ldots, n\}$ (supply nodes) and $N_2 = \{1', 2', \ldots, n'\}$ (demand nodes). There is an arc (i, j') for every $i \neq j$ with $u_{ij'} = 1$. We create the equivalent max-flow problem by adding nodes s and t, arcs (s, i) with $u_{si} = s_i$, and arcs (j', t) with $u_{j't} = (n-1) - s_{j'}$. If the solution of the max-flow problem saturates all the arcs going out of s (and hence all the arcs coming into t), the the scores are feasible.