Problem Set 1

Course: Algorithms for optimization and statistical inference (2014)

1. Show that every tree is a bipartite graph.

2. A *leaf* of a graph is a vertex v with degree one, $d_v = 1$. Prove that every tree with two or more nodes has at least two leaves. Characterize all trees with just two leaves.

3. Prove that for every tree $T = (V, E)$, $|E| = |V| - 1$ (Hint: what happens when you remove a leaf from a tree?)

4. Remember that if $G = (V_G, E_G)$ is connected, then there exists a tree $T = (V_T, E_T)$ such that $V_T = V_G$ and $E_T \subseteq E_G$ (i.e. every connected graph admits a spanning tree). Show that if $G = (V_G, E_G)$ is acyclic, then there exists a tree $T = (V_T, E_T)$ such that $V_T = V_G$ and $E_G \subseteq E_T$.

5. Using 3 and 4, show that the following are equivalent for $T = (V, E)$:

   (a) T is a tree.

   (b) T is connected, and $|E| \leq |V| - 1$.

   (c) T is acyclic, and $|E| \geq |V| - 1$.

6. Let $G = (V_G, E_G)$ be an undirected graph with edge weights $w : E_G \to (0, \infty)$. Consider $H = (V_H, E_H)$ a connected graph *of minimum weight* such that $E_F \subseteq E_G$ and $V_F = V_G$, that is, F is a *minimum weight spanning subgraph*. Prove that F is a minimum spanning tree of G.

7. How many bipartitions are there in a bipartite graph?

8. Given a weighted graph $G = (V, E)$, $w : E \to \mathbb{R}_+$ define $F = \{(i, j) \in E : w_{ij} < w_{ik} \forall k : (ik) \in E\}$. Let $T = (V, E')$ be a minimum spanning tree of G, w.

   (a) Show that $F \subset E'$.

   (b) Suppose that w is injective (i.e. all weights are different). Show that $|F| > |E'|/2$

9. Given a matrix A how can you compute A^k using only $O(\log k)$ matrix products? Show that the same procedure applies to the *min-sum* matrix product.

10. Prove that the number of spanning trees in the complete graph K_n is n^{n-2}.

11. Remember that given a weight matrix $W : \{1,\ldots,N\}^2 \to \mathbb{R}$, Kirchoff's matrix-tree theorem allows us to compute the "partition function" $Z_\beta = \sum_{T \text{ spanning tree}} e^{-\beta W(T)}$ where $W(T) = \sum_{(ij)\in E(T)} W(ij)$. This defines a Boltzmann probability distribution over spanning trees as follows:

$$P_\beta(T) = Z_\beta^{-1} e^{-\beta W(T)}$$

You can easily verify that this is a probability distribution. How can you compute a *marginal edge presence* probability

$$P_\beta(ij) = Z_\beta^{-1} \sum_{T \textit{ spanning tree:}(ij)\in E(T)} e^{-\beta W(T)}$$

12. How can you compute the average of W in P_β, i.e.:

$$\langle W \rangle = \sum_{T \text{spanning tree}} P_\beta(T) W(T)$$

13. Use the previous exercise to compute the *entropy* $S(P_\beta) = -\sum_{T \text{spanning tree}} P_\beta(T) \log P_\beta(T)$ of P_β. The entropy is a measure of how *broad* the distribution is. Show that for $\beta = 0$, the entropy reaches its maximum value, $S = (n-2)\log(n)$.

14. Let $A \in \mathbb{R}^{n\times n}$ be a matrix.

    (a) Devise an algorithm to find a permutation of rows, if it exists, leaving no zeros on the diagonal. That is, find $\pi : \{1,\ldots,n\} \to \{1,\ldots,n\}$ such that $A_{\pi(i),i} \neq 0$ for $i = 1,\ldots n$.

    (b) Show that π exists for any invertible matrix A

15. Consider *maximum flow problem with multiple sources and sinks* defined as follows: Given subsets $S, T \subset V$, and capacity function $c : V \times V \to \mathbb{R}_{\geq 0}$, we define a (c, S, T)-flow as $f : V \times V \to \mathbb{R}$ satisfying:

    (a) $f(u,v) \leq c(u,v)\ u,v \in V$

    (b) $f(u,v) = -f(v,u)\ u,v \in V$

    (c) $f(u,V) = 0\ \forall u \in V \setminus (S \cup T)$

    For such f we define $|f| = f(S,V) = f(V,T)$. Find a reduction of the problem of finding the maximum $|f|$ to a simple *max flow* problem.

16. Consider the problem defined as follows: Given a link capacity function $c : V \times V \to \mathbb{R}_{\geq 0}$, for source s and sink t and a *node capacity* function $d : V \to \mathbb{R}_{\geq 0}$, find the maximum (c, s, t)-flow $f : V \times V \to \mathbb{R}$ satisfying also $\frac{1}{2}\sum_{w\in V} |f(v,w)| \leq d(v)$ for $v \in V \setminus \{s,t\}$. Find a reduction of this problem to a simple *max-flow* problem.

17. Given an undirected graph G, a set of contacts $C \subseteq V$ and a set of exits $T \subset V$, find a set of non-overlapping paths (not even vertices are allowed to overlap) from each vertex in C to each vertex in T. That is, for each $c \in C$, we need a path $p^{(c)} = \left(c = p_1^{(c)}, p_2^{(c)}, \ldots, p_{k_c}^{(c)}\right)$ such that $p_i^{(c)} \neq p_j^{(d)}$ if $c \neq d$. Note: this problem is useful to deploy conductive tracks on a microchip. In this case the graph is a $2-d$ lattice.

18. Let $A \in \mathbb{R}^{n \times n}$ be a matrix such that $\sum_{j=1}^{n} A_{ij} \in \mathbb{N}$ for $i = 1, \ldots n$ and $\sum_{i=1}^{n} A_{ij} \in \mathbb{N}$ for $j = 1, \ldots n$. Find $B \in \mathbb{N}^{n \times n}$ such that

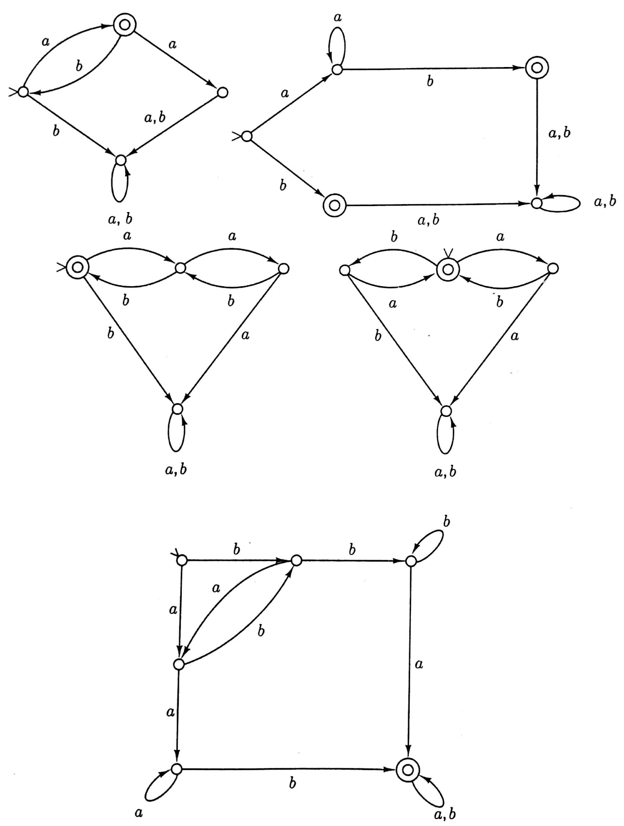
    (a) $|A_{ij} - B_{ij}| \leq 1$ for $i, j \in \{1, \ldots, n\}$

    (b) $\sum_{j=1}^{n} A_{ij} = \sum_{j=1}^{n} B_{ij}$ for $i = 1, \ldots n$.

    (c) $\sum_{i=1}^{n} A_{ij} = \sum_{i=1}^{n} B_{ij}$ for $j = 1, \ldots n$.

    Hint: Reduce to a flow problem, and think about the augmenting path algorithm.

19. Describe informally the languages accepted by the following deterministic finite automata:



20. Construct deterministic finite automata accepting the following languages:

    (a) $\{w \in \{a, b\}^* :$ each a in w is immediately preceded by a $b\}$

    (b) $\{w \in \{a, b\}^* : w$ has $abab$ as a substring$\}$

(c) $\{w \in \{a, b\}^* : w$ has neither aa nor bb as a substring$\}$

(d) $\{w \in \{a, b\}^* : w$ has an odd number of a's and an even number of b's$\}$

(e) $\{w \in \{a, b\}^* : w$ has both ab and baas a substring$\}$

(f) $\{w \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* : w$ correspond to a positive even integer$\}$

(g) $\{w \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* : w$ correspond to a positive integer divisible by 5$\}$

(h) $\{w \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* : w$ correspond to a positive integer divisible by 3$\}$

21. Given two deterministic finite automata deciding languages $L_1, L_2 \subset \Sigma^*$, build a new finite deterministic automata that decides $L_1 \cap L_2$.

22. Devise a (polynomial) method to: given a deterministic finite automata, find an input that would be accepted (if it exists) or to conclude that there is none (and thus that the automata decides the language $L = \emptyset$).

23. Given two deterministic finite automatas, devise a polynomial procedure to decide if they decide the same language.