COMP 251 2016, Assignment 4, due Monday December 5th 2016 23:59

1. Maximum Deadline

28. Recall the scheduling problem from Section 4.2 in which we sought to minimize the maximum lateness. There are n jobs, each with a deadline d_i and a required processing time t_i, and all jobs are available to be scheduled starting at time s. For a job i to be done, it needs to be assigned a period from $s_i \geq s$ to $f_i = s_i + t_i$, and different jobs should be assigned nonoverlapping intervals. As usual, such an assignment of times will be called a *schedule*.

In this problem, we consider the same setup, but want to optimize a different objective. In particular, we consider the case in which each job must either be done by its deadline or not at all. We'll say that a subset J of the jobs is *schedulable* if there is a schedule for the jobs in J so that each of them finishes by its deadline. Your problem is to select a schedulable subset of maximum possible size and give a schedule for this subset that allows each job to finish by its deadline.

[10%] (a) Prove that there is an optimal solution J (i.e., a schedulable set of maximum size) in which the jobs in J are scheduled in increasing order of their deadlines.

[10%] (b) Assume that all deadlines d_i and required times t_i are integers. Give an algorithm to find an optimal solution. Your algorithm should run in time polynomial in the number of jobs n, and the maximum deadline $D = \max_i d_i$.

2. Critical edges

[5%] 7.21. An edge of a flow network is called *critical* if decreasing the capacity of this edge results in a decrease in the maximum flow. Give an efficient algorithm that finds a critical edge in a network.

[5%] 7.22. In a particular network $G = (V, E)$ whose edges have integer capacities c_e, we have already found the maximum flow f from node s to node t. However, we now find out that one of the capacity values we used was wrong: for edge (u, v) we used c_{uv} whereas it should have been $c_{uv} - 1$. This is unfortunate because the flow f uses that particular edge at full capacity: $f_{uv} = c_{uv}$.

We could redo the flow computation from scratch, but there's a faster way. Show how a new optimal flow can be computed in $O(|V| + |E|)$ time.

[10%] 7.23. A *vertex cover* of an undirected graph $G = (V, E)$ is a subset of the vertices which touches every edge—that is, a subset $S \subset V$ such that for each edge $\{u, v\} \in E$, one or both of u, v are in S.

Show that the problem of finding the minimum vertex cover in a *bipartite* graph reduces to maximum flow. (*Hint:* Can you relate this problem to the minimum cut in an appropriate network?)

3. Blood Bank

8. Statistically, the arrival of spring typically results in increased accidents and increased need for emergency medical treatment, which often requires blood transfusions. Consider the problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient.

   The basic rule for blood donation is the following. A person's own blood supply has certain *antigens* present (we can think of antigens as a kind of molecular signature); and a person cannot receive blood with a particular antigen if their own blood does not have this antigen present. Concretely, this principle underpins the division of blood into four *types*: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus, patients with type A can receive only blood types A or O in a transfusion, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.[4]

[10%]

   (a) Let s_O, s_A, s_B, and s_{AB} denote the supply in whole units of the different blood types on hand. Assume that the hospital knows the projected demand for each blood type d_O, d_A, d_B, and d_{AB} for the coming week. Give a polynomial-time algorithm to evaluate if the blood on hand would suffice for the projected need.

[10%]

   (b) Consider the following example. Over the next week, they expect to need at most 100 units of blood. The typical distribution of blood types in U.S. patients is roughly 45 percent type O, 42 percent type A, 10 percent type B, and 3 percent type AB. The hospital wants to know if the blood supply it has on hand would be enough if 100 patients arrive with the expected type distribution. There is a total of 105 units of blood on hand. The table below gives these demands,

blood type	supply	demand
O	50	45
A	36	42
B	11	10
AB	8	3

   Is the 105 units of blood on hand enough to satisfy the 100 units of demand? Find an allocation that satisfies the maximum possible number of patients. Use an argument based on a minimum-capacity cut to show why not all patients can receive blood. Also, provide an explanation for this fact that would be understandable to the clinic administrators, who have not taken a course on algorithms. (So, for example, this explanation should not involve the words *flow*, *cut*, or *graph* in the sense we use them in this book.)

4. Escape Problem (26-1)

An $n \times n$ ***grid*** is an undirected graph consisting of n rows and n columns of vertices, as shown in Figure 26.11. We denote the vertex in the i^{th} row and the j^{th} column by (i, j). All vertices in a grid have exactly four neighbours, except for the boundary vertices, which are the points (i, j) such that $i = 1$, $i = n$, $j = 1$, or $j = n$.

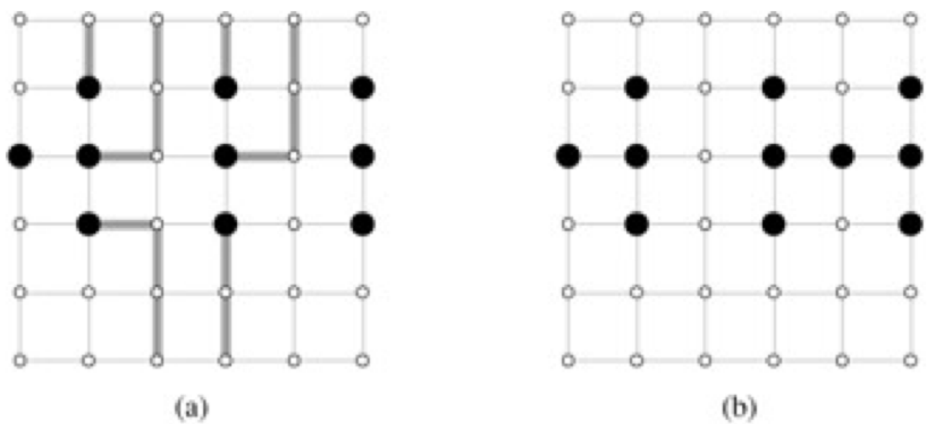

Figure 26.11: Grids for the escape problem. Starting points are black, and other grid vertices are white. *(a)* A grid with an escape, shown by shaded paths. *(b)* A grid with no escape.

Given $m \leq n^2$ starting points $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ in the grid, the ***escape problem*** is to determine whether or not there are m vertex-disjoint paths from the starting points to any m different points on the boundary. For example, the grid in Figure 26.11(a) has an escape, but the grid in Figure 26.11(b) does not.

[10%] **a)** Consider a flow network in which vertices, as well as edges, have capacities. That is, the total positive flow entering any given vertex is subject to a capacity constraint. Show that determining the maximum flow in a network with edge and vertex capacities can be reduced to an ordinary maximum-flow problem on a flow network of comparable size.

[10%] **b)** Describe an efficient algorithm to solve the escape problem, and analyze its running time.

5. CFL

In the Canadian Football League, teams can either win a game (2 points), tie a game (1 point) or lose a game (0 point). The table below shows the East division of the CFL at some point in time. The columns are explained as caption under the table. Notice that R > O+H+M+T because each team has more games to play against other teams outside the East division.

The question you should answer is wether Toronto is currently eliminated or not as candidate for finishing with the largest number of points in its division.

[10%] **a)** Write the general formulation of this situation as a Max Flow problem as we did in class for baseball.

[5%] **b)** Give a general formula for exhibiting that a team is eliminated (a certificate of elimination).

[5%] **c)** Show that Toronto is now eliminated.

• teams •	GP	W	L	T	pts	R	Details of R O	H	M	T
Ottawa	18	9	8	1	19	12	•	2	3	3
Hamilton	18	7	11	0	14	12	2	•	5	2
Montréal	18	7	11	0	14	12	3	5	•	1
Toronto	24	5	19	0	10	6	3	2	1	•

GP = Games Played
W = Wins
L = Losses
T = Ties
pts = Points = 2W+T
R = Games Remaining