Final Exam Review Problem Set
CS561
Fall 2014

Aaron Gonzales

December 4, 2014

1 CLRS 26-3, Algorithmic Consluting

Professor Gore wants to open up an algorithmic consulting company. He has identified n important subareas of algorithms which he represents by the set $A = \{A_1, A_2, \ldots A_n\}$ In each subarea A_k he can hire an expert in that area for c_k dollars. The consulting company has lined up a set $J = (J_1, J - 2 \ldots J_m)$ of potential jobs. In order to perform job J_i the company needs to have hried experts in a subset $R_i \subseteq A$ of subareas. Each expert can work on multiple jobs simulaneously. If the company chooses to accept job J_i, it must have hired eperts in all subares in R_i, and it will take in revenue of p_I dollars.

Professor Gore's job is to determine which subareas to hire experts in and hwich jobs to accept in order to maimize the net revenue, which is the total income from jobs accepted minus the total cost of employing the experts.

Consider the following flow network G. It contains a source vertex s, vertices $A_1, A_2, \ldots A_n$, vertices $J_1, J_2, \ldots J_m$ and a sink vertex t. for $k = 1, 2, \ldots n$, the flow network contains an edge $s < A_k$ with capacity $c(s, A_k) = c_k$, and for $i = 1, 2, \ldots m$, the flow network contains an edge (J_i, t) with capacity $c(J_i, t) = p_i$ For $k = 1, 2, \ldots n$ and $i = 1, 2, \ldots m$ if $A_k \in R_i$ then G contains an edge $A_k, J_i)$ with capacity $c(A_k, J_i) = \infty$.

(a) Show that if $J_i \in T$ for a finite-capacity cut (S, T) of G, then $A_k \in T$ for each $A_k \in R_i$.

(b) Show how to determine the maxium net revenue from the capacity of a minmum cut of G and the given p_i values.

(c) Give an efficient algorithm to determine which jobs to accept and which experts to hire. Analyze the running time of your algorithm in terms of m, n, and $r = \sum_{i=1}^{m} |R_i|$.

2 34.5-2; 0-1 integer programming

Given an integer $m \times n$ matrix A and an integer m-vector b, the *0-1 integer-programming problem* asks whether there exists an integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$. Prove that 0-1 integer programming is NP-Complete. (Hint: Reduce from 3-CNF-SAT)

3 problem 1 from lect 23 erickson

Suppose you are given a directed graph $G = (V, E)$, two vertices s and t, a capacity function $c : E \to \mathbf{R}^+$, and a second function $F : E \to \mathbb{R}$. Describe an algorithm to determine wether f is a maximum (s, t) flow in G.

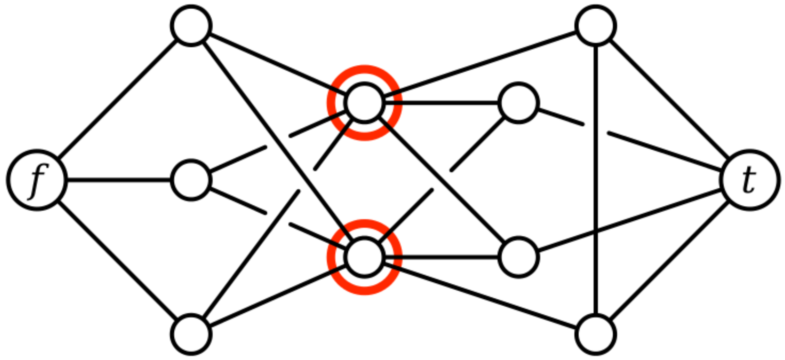


Figure 1: Graph for problem 5 from erickson's notes

4 problem 5 from 24, erickson

The Island of Sodor is home to a large number of towns and villiages, connected by an extensive rail network. Recently, several cases of a deadly contagious disease (either swine flu or zombies; reports are unclear) have been reported in the village of Ffarquhar. The controller of the Sodor railway plans to close down certain railway stations to prevent the disease from spreading to Tidmouth, his home town. No trains can pass through a closed station. To minimize expense (and public notice), he wants to close down as few stations as possible. However, he cannot close the Ffarquhar station, because that would expose him to the disease, and he cannot close the tidmouth station, because then he couldn't visit his favorite pub.

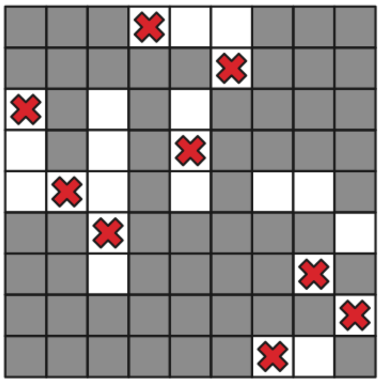
Describe and analyze an algorithm to find the minimum number of stations that must be closed to block all rail travel from Ffarquhar to Tidmouth. The Sodor rail network is represented by an undirected graph, with a vertex for each station and an edge for each rail connection between two stations. Two special vertices f and t represent the stations in Ffarquhar and Tidmouth.

For example, given the following input graph (Figure 1), your algorithm should return the number 2.

5 problem 8 from 24, erickson

Suppose we are given an $n \times n$ square grid, some of whose squares are colored black and the rest white. Describe and analyze an algorithm to determine whether tokens can be placed on the grid so that

- **every token is on a white square;**
- **every row of the grid contains exactly one token; and**
- **every column of the grid contains exactly one token**



Your input is a two-dimensional array *IsWhite*$[1..n, 1..n]$ of booleans, indicating which squares are white. Your output is a single boolean. For example, given the grid above as input, your algorithm should return TRUE.

6 34-2 CLRS

Bonnie and Clyde have just robbed a bank. They have a bag onf money and want to divide it up. For each of the following scenarios, either give a polynomial-item algorithm, or prove that the problem is NP-Complete. The input in each case is a list of the n items in the bag, along with the value of each.

Jared's note:For this problem, it may be helpful to use the fact that SUBSET-SUM is NP-Hard. See Section 34.5.5 for a defintion of SUBSET-SUM. (You only need to read the definition in this section, not the reduction that proves that SUBSET-SUM is NP-Hard.

(a) **The bag contains n coins, but only 2 different denominations: some coins are worth x dollars, and some are worth y dollars. Bonnie and Clyde wish to divide the money exactly evenly.**

(b) **The bag contains n coins, with an arbitrary number of diferent denominations, but each denomination is a nonnegative integer power of 2, ie.e, the possible denominations are 1 dollar, 2 dollars, 4, dollars, etc. Bonnie and Clyde wish to divide the money exactly evenly.**

(c) The bag contains n checks, which are, in an amazing coincidence, made out to "Bonnie or Clyde." They wish to divide the checks to that they each get the exact same amount of money.

(d) The bag contains n checks as in part (c), but this time bonnie and Clyde are willing to accept a split in which the difference is no larger than 100 dollars.

7 Problem 34-4 from CLRS

Suppose that we have one machine and a set of n tasks $a_1, a_2, \ldots a_n$, each of which requires time on the machine. Each task a_j requires t_j time units on the machine (its processing time), yields a prfit of p_j, and has a deadline d_j. The machine can process only one task at a time, and task a_j must run without interrupotion for t_j consecutive time units. If we complete task a_j by its deadline d_j, we receive a profit p_j but if we complete it after its deadline, we receive no profit. As an optimization problem, we are given the rpocessing times, profits, and deadlines for a set of n tasks, and we wish to find a schedule that completes all the tasks and returns the greatest amount of profit. The processing times, profits, and deadlines are all nonnegative numbers.

(a) State this problem as a decision problem.

(b) Show that the decision problem is NP-Complete

(c) Give a polynomial-time algorithm for the decision problem, assuming that all processing times are integers from 1 to n. (Hint: use dynamic programming)

(d) give a polynomial-time algoirthm for the optimization problem, assuming that all processing times are integers from 1 to n.