CSI 4105 Design and Analysis of Algorithms II — Winter 2012
Computer Science — University of Ottawa

Homework Assignment #3 (100 points, weight 6.67%)
Due: Friday Mar 30, at 11:30 p.m. (in lecture)

Internet-free answers required. I have inserted enough hints so that inspiration should not be sought anywhere else. I warn you that several internet sources for some of these questions are either not detailed enough or have subtle mistakes. Any answers similar to external sources (other than your mind or textbook) will be severely penalized.

1. (35 points) **Bin Packing Approximation Algorithm**

   Suppose that we are given a set of n objects, where the size of the ith object satisfies $0 < s_i \leq 1$. We want to pack the objects into the minimum number of bins of size 1. Each bin can hold any subset of the objects whose total size does not exceed 1. You know the decision version of boin-packing is NP-complete (proven in the midterm test 2).

   The **first-fit** heuristic takes each object in turn and places it into the **first** bin that can accommodate it. The following subquestions are steps towards showing that the first fit heuristic is a 2-approximation algorithm for the problem.

   Let $S = \sum_1^n s_i$.

   (a) (5 marks) Prove that the optimal number of bins required is at least $\lceil S \rceil$.

   (b) (5 marks) Prove that the first-fit heuristic leaves at most one bin less than half full.

   (c) (7 marks) Prove that the number of bins used by the first-fit heuristic is never more than $\lceil 2S \rceil$.

   (d) (8 marks) Prove that the first-fit heuristic is an approximation algorithm with an approximation ratio of 2.

   (e) (10 marks) Application. At your co-op term you are asked to minimize the number of trucks to transport a variety of heavy items. Space is not an issue, as items are so heavy that you would exceed weight capacity before the space is full. Each truck has weight capacity 1000Kg, and each of the n items has a weight w_i. Your boss requires that you design a polynomial time algorithm to solve this problem to optimality in 3 weeks, and says that failing to do so would affect his evaluation of your co-op report. After spending one week trying to find a polynomial time algorithm, you remember your CSI4105 and are thankful that you can argue with your boss that his request is not reasonable. However, you explain not all is lost and you are able to design an approximation algorithm for the problem. The next steps outline your explanation:

   - Give a brief explanation of why the truck problem is NP-hard.
   - Show how to use the 2-approximation algorithm for bin-packing in order to solve the truck problem.

2. (30 marks) **Bin Packing Exact Algorithm**

   Write a backtracking algorithm to solve bin-packing optimally (see problem in the previous question); the solution of the problem should yield the minimum number of bins. The algorithm should use polynomial space; the algorithm is not expected to run in polynomial time, since the decision version of the problem is NP-complete (this was hopefully proven by you in the midterm test 2). Justify its **polynomial space** and **analyse its running time** by giving an upper bound on it.

3. (35 points) **Competitive Facility Location**

   Recall that in Competitive Facility Location, we are given a graph G with a non-negative value b_i attached to each node i, and a bound B. Two players alternately select vertices of G, so that the set of selected vertices at all times forms an independent set (can't select a neighbour of an already selected node). Player 2 wins if he ultimately selects a set of vertices of total value at least B; Player 1 wins if he prevents this from happening.

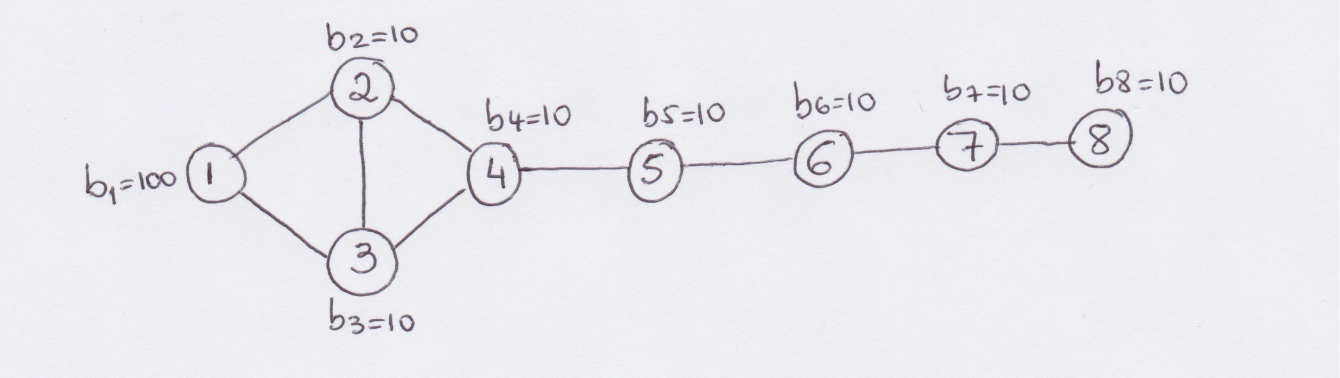
   The question is: Given the graph G with values b_i attached to each node i and given the bound B, is there a strategy by which Player 2 can force a win?

   Consider the example below.

   Two coffee houses play this game: player 1 is GIMEBUCK$ and player 2 is BRIDGEFAIR. In the graph below, the answer is **Yes** for $B = 20$. To verify that BRIDGEFAIR is guaranteed to be able to set up facilities for a total value of at least 20, we need to analyse some cases:

   1) Case 1: player 1 GIMEBUCK$ picks one of vertices 2,3,4,5,6,7,8.
   In this case, BRIDGEFAIR wins by picking vertex 1, yielding value 100.

   2) Case 2: player 1 GIMEBUCK$ picks vertex 1.
   BRIDGEFAIR can now pick any of vertices 4,5,6,7,8; the winning strategy will be to pick vertex 6, earning value 10 this round, and forcing GIMEBUCK$ to pick one of vertices 4 or 8. In either case, BRIDGEFAIR can then pick the other unpicked vertex (8 or 4, respectively), and accumulate 10 more in value, for a total value of 20.



   Show that Competitive Facility Location is in **PSPACE**, by giving an algorithm (pseudo code) that solves the problem in a **polynomial amount of space**. Justify why the space used is polynomial.