To receive credit for a problem, you must both have sufficient narrative description of your algorithm and a sentence stating what the intended meaning of each array entry is (e.g. $A(i, j)$ is 1 if there is a subset of the first i items with aggregate sum j).

20. (2 points) Consider the problem where the input is a collection of n train trips within Germany. For the ith trip T_i you are given the date d_i of that trip, and the non-discounted fare f_i for that trip. The German railway system sells a Bahncard for B Euros that entitles you to a 50% fare reduction on all train travel within Germany within the next L days of purchase. The problem is to determine when to buy a Bahncard to minimize the total cost of your travel.

    For example if the input was d_1 = January 11, 1997, $f_1 = 20$ Marks, d_2 = February 11, 1998, $f_2 = 200$ Marks, d_3 = January 11, 1999, $f_3 = 200$ Marks, d_4 = March 13, 1999, $f_4 = 100$ Marks, d_5 = February 11, 2002, $f_5 = 200$ Marks, and d_6 = January 11, 2003, $f_6 = 600$ Marks, $B = 240$, and $L = 365$ then you might buy a Bahncard on February 11, 1998, and February 11, 2002, resulting in a total cost of 1200 Marks.

    Give a polynomial-time dynamic programming algorithm for this problem. The running time of you algorithm should be independent of B and L.

21. (4 points) Assume that you are given a collection $B_1, \ldots, B_n$ of boxes. You are told that the weight in kilograms of each box is an integer between 1 and some constant L, inclusive. However, you do not know the specific weight of any box, and you do not know the specific value of L. You are also given a pan balance. A pan balance functions in the following manner. You can give the pan balance any two disjoint sub-collections, say S_1 and S_2, of the boxes. Let $|S_1|$ and $|S_2|$ be the cumulative weight of the boxes in S_1 and S_2, respectively. The pan balance then determines whether $|S_1| < |S_2|$, $|S_1| = |S_2|$, or $|S_1| > |S_2|$. You have nothing else at your disposal other than these n boxes and the pan balance. The problem is to determine if one can partition the boxes into two disjoint sub-collections of equal weight. Give an algorithm for this problem that makes at most $O(n^2\ L)$ uses of the pan balance. For partial credit, find an algorithm where the number of uses is polynomial in n and L.

22. (6 points) The input to this problem is n points $x_1, \ldots, x_n$ on a line. A good path P has the property that one endpoint of P is the origin and every x_i is covered by P. Note that P need not be simple, that is, it can backtrack over territory that it has already covered. Assume a vehicle moves along this path from the origin at unit speed. The response time r_i for each x_i is the time until the vehicle first reaches x_i. The problem is to find the good path that minimizes $\sum_{i=1}^{n} r_i/n$, the average response time. For example, if the points are $x_1 = 1$ $x_2 = 8$ and $x_3 = -2$ and the path visited the points in the order x_1, x_3, x_2, the average response time for this path would be $1/3 + (1 + 3)/3 + (1 + 3 + 10)/3$. Give a polynomial-time dynamic programming algorithm for this problem.