Recitation 6

The aim of this recitation is to apply dynamic programming techniques to compute (after the fact) an optimal betting strategy for a pool to wager on the outcome of the 2001 NCAA men's basketball tournament.

Name and NetID:
Section:

1 Introduction

For those unfamiliar with this tournament, 64 teams compete in a single elimination bracket consisting of 6 rounds. I each game, the winner advances and the loser is eliminated. (There are 32 winners that advance to round 2, 16 to round 3, 8 to round 4, etc.) So there are 63 games total, and a team must win 6 games in a row to become the champion.

The bracket is split into four regions of 16 teams apiece, and each team is given a seed from 1 to 16 (where this "seed"ing is a hypothesized ranking of these 16 teams from best to worst). The teams that have played the best during the season are rewarded with the lower numbered seeds, so the four #1 seeds are the favorites to win, while the four # 16 seeds are perceived to have little chance of winning.

The following novel pool was invented by Robin Lock and popularized by former Cornell ORIE faculty member Rick Cleary. The teams are each given a price based on their seed, according to the table below. The costs are given in cents.

seed	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cost	25	21	18	15	12	10	8	6	5	4	3	2	1	1	1	1

Before the start of the tournament, each player who enters the pool is allowed to buy up to one dollar's worth of teams. For instance, one could buy all four # 1 seeds for 25 cents apiece. Or one could buy three # 1 seeds, one # 3 seed, one # 11 seed, and all four # 16 seeds. At the end of the tournament, each player adds up the total number of games won by teams she purchased, and the player with the highest total wins the pool.

To figure out the best possible entry before the tournament starts, takes knowledge, skill, and a good crystal ball. But to figure out, after the fact, what the best possible entry *would have been*, takes dynamic programming. The table below gives the relevant data from the actual 2001 tournament. It lists each of the 32 teams who won at least one game, along with the number of games the team won and its seed. What would the optimal entry have been for the 2001 tournament pool? As an example, one could buy Duke, Arizona, Maryland, Michigan St., and USC. This selection would cost 99 cents and yield 22 wins.

Team	Wins	Seed
Michigan St	4	1
Fresno St	1	9
Gonzaga	2	12
Indiana St	1	13
Temple	3	11
Florida	1	3
Penn St	2	7
North Carolina	1	2
Illinois	3	1
Charlotte	1	9
Syracuse	2	5
Kansas	2	4
Notre Dame	1	6
Mississippi	2	3
Butler	1	10
Arizona	5	2

Team	Wins	Seed
Duke	6	1
Missouri	2	9
Utah St	1	12
UCLA	2	4
USC	3	6
Boston College	2	3
Iowa	1	7
Kentucky	2	2
Stanford	3	1
St. Joseph's	1	9
Cincinnati	2	5
Kent St	1	13
Georgia St	1	11
Maryland	4	3
Georgetown	2	10
Hampton	1	15

2 Dynamic programming formulation

First, you will be led through a numer of steps that cast this optimization problem as a dynamic program. The aim is to decide, for each team, how much of your total budget of 100 pennies that you wish to bet on that team.

1. To what do the decision stages correspond? Recall that in determining the stages, you might first think about how you would specify any feasible solution (see above).

2. What is the appropriate state information? (That is, given a sequence of decision in the first 12 stages, what collective information about all of those decisions do you *really* need to know in order to merge those decisions with an optimal set of decisions for the remaining stages?)

3. What options (transitions) are possible for a given stage in a given state? (As a warmup, you might think of the options that are possible for the first stage, starting from your initial state.)

4. What is the optimization function that corresponds to each state in each stage of the problem? Express this function just by giving an English-language description.

5. What is the dynamic programming recurence for this optimization function?

6. What are the base conditions to start the dynamic programming computation? (Hint: you might want to set up a dummy stage that corresponds to the point where all decisions have already been made.)

7. The next phase of this exercise is to solve this problem using AMPL. As a warmup, use `inventory.mod` and `inventory.dat` files (on Blackboard, under "Recitations") to determine the optimal value and the optimal solution for that give input. What are they?

8. In setting up an AMPL implementation of your formulation of the betting problem, you will need to set up both a .mod file and a .dat file. Your data file should look something like the two tables given here (and essentially nothing more, other than a parameter specifying the number of teams involved). To save you some typing effort, a text file with the two tables has been posted: NCAATables.txt (on Blackboard, under "Recitations"). The conversion of seeds to costs should be done inside the model. List the teams you chose in your optimal entry, along with the total number of wins you have and your total cost.