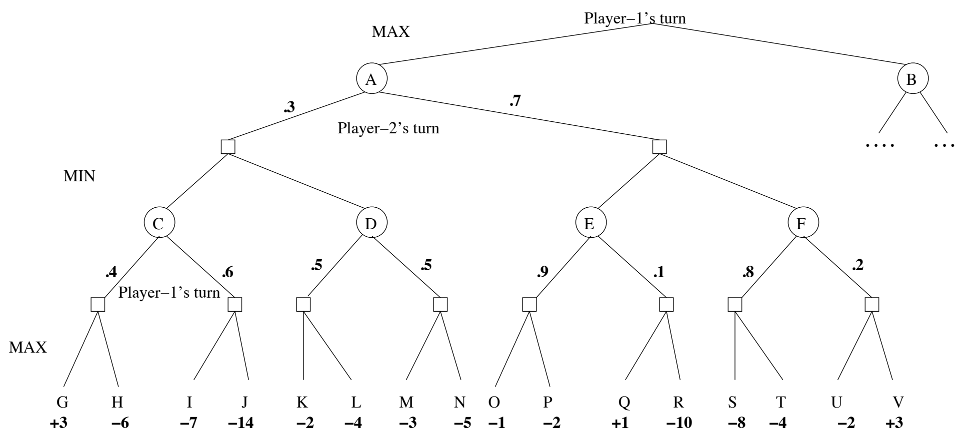
C2 Artificial Intelligence (25 points)

Games, Uncertainty. (9 points) The minimax algorithm is used to find the best move in a game tree. Suppose however that the game involves an element of chance, in which a player's possible moves depend on the throw of a dice, the spin of a wheel, or some other phenomenon whose result is uncertain.

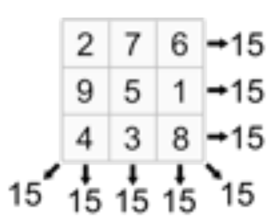


Consider the above game tree, in which Player-1 (who wants to maximize the value of the resulting state of the game) can choose to move to A or to B. At this point, Player-2's possible moves are uncertain and depend on a chance event, as indicated by the labelled chance nodes (shown as circles in the above game tree)—for example, from A, the probability is .3 that Player-2 can choose to move to C or to D, and the probability is .7 that Player-2 can choose to move to E or to F. Similarly, if Player-2 moves to C, the probability is .4 that Player-1 can choose to move to G or H, and the probability is .6 that Player-1 can choose to move to I or J.

Given this game tree, determine what the backed-up value of node B must be if the best decision for Player-1 is to move to A. You must explain your answer in detail.

Search. (16 points) In recreational mathematics, a **magic square** of order *n* is an arrangement of n^2 numbers, usually distinct integers, in a square, such that the *n* numbers in all rows, all columns, and both diagonals sum to the same constant. A **normal** magic square contains the integers from *1* to n^2.

Normal magic squares exist for all orders n ≥ 1 except n = 2, although the case n = 1 is trivial, consisting of a single cell containing the number 1. The smallest nontrivial case, shown below, is of order 3.



The constant sum in every row, column and diagonal is called the magic constant or magic sum, M. The magic constant of a normal magic square depends only on *n* and has the value

$$M = \frac{n(n^2+1)}{2}.$$

(a) [6 pts] Formulate the construction of a normal magic square of size n = 3 as a constraint satisfaction problem. Specify what would be taken as variables, and (informally) specify the constraints on those variables.

(b) [10 pts] Set up the problem and show the first steps of processing using backtracking with forward checking. Show the assignments and variable domains through either 4 assignments or until the algorithm needs to backtrack – whichever comes first.