Consider the zero-sum expectimax game tree from Section 3 shown below. Circles represent chance nodes. Trapezoids that point up represent choices for the player seeking to maximize. Outcome values for the maximizing player are listed for each leaf node.

(a) First, assume that each chance node chooses uniformly between available moves. Assuming optimal play, carry out the expectimax search algorithm and write the value of each node inside the corresponding trapezoid. What is the expected value of this game assuming optimal play? What is the optimal move to make at each node?

![Game Tree](image)

The game is worth 3. We should make the move that takes us left.

(b) Now, assume that each chance node plays the leftmost move with probability 0.5, the middle move with probability 0.25, and the rightmost move with probability 0.25. Assuming optimal play, what is the expected value of this game? What is the optimal move to make?

![Game Tree](image)

This game is still worth 3. But the optimal move to make is now the center path.
2 Non-transitive Dice

Consider the following three 4-sided dice A, B, C with the given side values. Assume the dice are all fair, and all rolls are independent.

A: 3,3,3,7
B: 2,2,5,7
C: 1,4,5,6

(a) What is the expected value of each die?

A: \[
\frac{1}{4}(3 + 3 + 3 + 7) = 4
\]

B: \[
\frac{1}{4}(2 + 2 + 5 + 7) = 4
\]

C: \[
\frac{1}{4}(1 + 4 + 5 + 6) = 4
\]

(b) Consider the indicator function \(\text{better}(X, Y)\) which has value 1 if \(X > Y\), -1 if \(X < Y\), and value 0 o.w. What are the expected values of \(\text{better}(A, B)\), \(\text{better}(B, C)\), and \(\text{better}(C, A)\)?

For \(\text{better}(A, B)\), we compute the expectation by exhaustively enumerating all values of \(A, B\). Note that since the dice are fair and independent, \(P(A = a_1, B = b_1) = P(A = a_1)P(B = b_1) = \frac{1}{16}\).

\[
E[\text{better}(A, B)] = \sum_{a_i \in A} \sum_{b_i \in B} P(A = a_i, B = b_i) \text{better}(A = a_i, B = b_i)
\]

\[
= \frac{1}{16}((\text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 5) + \text{better}(A = 3, B = 7)) + \\
\text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 5) + \text{better}(A = 3, B = 7)) + \\
\text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 2) + \text{better}(A = 3, B = 5) + \text{better}(A = 3, B = 7)) + \\
(\text{better}(A = 7, B = 2) + \text{better}(A = 7, B = 2) + \text{better}(A = 7, B = 5) + \text{better}(A = 7, B = 7)))
\]

\[
= \frac{1}{16}((1 + 1 + -1 + -1) + (1 + 1 + -1 + -1) + (1 + 1 + -1 + -1) + (1 + 1 + 1 + 0))
\]

\[
= \frac{3}{16}
\]

Similarly, for \(\text{better}(B, C)\), we have

\[
E[\text{better}(B, C)] = \frac{1}{16}((1 + -1 + -1 + -1) + (1 + -1 + -1 + -1) + (1 + 1 + 0 + -1) + (1 + 1 + 1 + 1)) = \frac{1}{16}
\]

Finally, for \(\text{better}(C, A)\), we have

\[
E[\text{better}(C, A)] = \frac{1}{16}((-1 + -1 + -1 + -1) + (1 + 1 + -1 + -1) + (1 + 1 + 1 + -1) + (1 + 1 + 1 + -1)) = \frac{2}{16} = \frac{1}{8}
\]

A binary relation \(R\) is transitive if for all \(a, b, c\), \(R(a, b)\) and \(R(b, c)\) implies \(R(a, c)\). For example, the greater-than operator \(\langle\rangle\) on the real numbers is an example of a transitive relation: if \(x < y\) and \(y < z\) then \(x < z\).

(c) Why are dice like those in (a) sometimes called “non-transitive dice”?

We generally think of better as a transitive operator: if \(A\) is better than \(B\) and \(B\) is better than \(C\), \(A\) should be better than \(C\). However, this is not the case for the three dice given above; thus, they are “non-transitive”.
3 St. Petersburg Paradox

You are traveling in Russia when a friendly stranger comes up to you and offers you the following game. He will give you $1 and a fair coin. You flip the coin repeatedly until the first tail appears. For every head that appears, you double your money. When the first tail appears, the game is over.

(a) What is the expected value of this game?

There is a \( \frac{1}{2} \) probability of losing immediately for a payoff of $1. There is a \( \frac{1}{2^2} \) probability of losing after HT (one head and then a tail) for a payoff of $2. In the general case, there is a \( \frac{1}{2^n+1} \) probability of losing after seeing \( n \) heads and 1 tail, for a payoff of \( 2^n \).

The expected value of this game is then:

\[
\frac{1}{2}(1) + \frac{1}{4}(2) + \cdots + \frac{1}{2^{n+1}}(2^n) + \cdots = \frac{1}{2} + \frac{1}{2} + \cdots + \frac{1}{2} = \infty
\]

(b) What would you pay to play this game? How does this compare to your answer from part (a)?

Up to you: probably no more than $2-$10, certainly not infinite.

This game is known as the St. Petersburg Paradox. One way of resolving this paradox is by using the decreasing marginal utility of money.

(c) Assume that you have a utility function for money of the form \( U(x) = \log_2(x) \) where \( x \) is your total winnings from this game. What is the expected utility of this game to you? What is the expected value of this game in dollars? You may find the following identities helpful:

\[
r + r^2 + r^3 + \cdots = \sum_{i=0}^{\infty} r^{i+1} = \frac{r}{1-r}, \quad r \in [0,1)
\]

\[
r(0) + r^2(1) + r^3(2) + r^4(3) + \cdots = \sum_{i=0}^{\infty} ir^{i+1} = \left(\frac{r}{1-r}\right)^2, \quad r \in [0,1)
\]

The payoffs after 1, 2, ..., \( n \) heads are \( \log_2(2^0), \log_2(2^1), \ldots, \log_2(2^n) = 0, 1, \ldots, n \), respectively. We can then compute expected utility as

\[
S = \frac{1}{2}(0) + \frac{1}{4}(1) + \frac{1}{8}(2) + \cdots
\]

We can compute the value of the above sequence using the trick of multiplying \( S \) by \( \frac{1}{2} \) and subtracting it from the original

\[
S = \frac{1}{2}(0) + \frac{1}{4}(1) + \frac{1}{8}(2) + \cdots
\]

\[
-\frac{1}{2}S = -\frac{1}{4}(0) - \frac{1}{8}(1) - \cdots
\]

\[
\Rightarrow S - \frac{1}{2}S = \frac{1}{4}(1) + \frac{1}{8}(1) + \cdots = \frac{1}{1 - \frac{1}{2}}
\]

\[
\Rightarrow S = 1
\]
Note that this is non-infinite! The equivalent dollar amount is $2^1 = 2$. 
4 Suicidal Pacman

Pacman is sometimes suicidal when doing a minimax search because of its worst case analysis. We will build here a small expectimax tree to see the difference in behavior.

Consider the following rules (slightly simplified from assignment):

- Ghosts cannot change direction unless they are facing a wall. The possible actions are east, west, south, and north (not stop). Initially, they have no direction and can move to any adjacent square.
- We use random ghosts which choose uniformly between all their legal moves.
- Assume that Pacman cannot stop
- If Pacman runs into a space with a ghost, it dies before having the chance to eat any food which was there.
- The game is scored as follows:
  - -1 for each action Pacman takes
  - 10 for each food dot eaten
  - -500 for losing (if Pacman is eaten)
  - 500 for winning (all food dots eaten)

Given the following “trapped” maze, build the expectimax tree with max and chance nodes clearly identified. Use the game score as the evaluation function at the leaves. If you don’t want to make little drawings, all possible states of the game have been labeled for you on the next page: use them to identify the states of the game. Pacman moves first, followed by the lower left ghost, then the top right ghost.
(a) Build the expectimax tree. What is Pacman’s optimal move? Play W for an expected payoff of 3.

(b) What would Pacman do if it was using minimax instead?
If we treat the ghost nodes as minimizing nodes and run minimax, we see that if Pacman plays W the ghosts would play N,E respectively, and we would be stuck with a payoff of -502.

Instead, we could earn a better payoff of -501 by immediately playing E: suicidal Pacman!

(c) By changing the probabilities of action for the ghosts, can you get expectimax to make the same decision as minimax?

One possible choice is for the ghosts to play N 99.95% of the time if N is legal and to choose randomly among remaining legal moves the rest of the time. Then at the first chance node for the blue ghost, we play N 99.95% of the time and E 0.05% of the time, and the corresponding payoff is $0.9995 \times (-502) + 0.0005 \times (508) = -501.5$, which is worse than playing E immediately for a payoff of -501.

(d) Now say you are using the following alternate game score components:

-1 for Pacman making a move
-1.5 for losing
0 for eating food
0.3 for winning

Use this new game score as your evaluation function at the leaves. Note this yields a monotonic transformation of the original utilities: a function which preserves the ordering of the state according to their utility. Could this change the decision of Pacman using expectimax?

The scores at the leaves (reading left to right) are -3.5, -3.5, -1.7, -3.5, -2.5. Propagating up the tree, we see that Pacman gets expected value $0.5 \times (-1.7) + 0.5 \times (-3.5) = -2.6$ from playing W and $-2.5$ from playing E, and we see that his optimal decision has changed. Optimal decisions are sensitive to monotonic transformations because the probabilities involved are not scaled.