STATE INDEPENDENT STOPPING PROBLEMS.

In each round of a game, a die is rolled. If the roll is a 1, the game stops and you get nothing. Otherwise you may choose to continue (and collect nothing at this time) or stop and collect $1 for each dot that turns up. When should you stop?

This is a state independent stopping problem. The probability of the next state is independent of the current state.
Suppose we have a state independent problem with $m+1$ states \{0, 1, ..., m\} and suppose 0 is the dead state $\Delta$.

For the dead state, the transition probabilities are $T(0, 0) = 1$, $T(0, j) = 0$ otherwise.

For all other states, the probability $p_j$ of transition to a next state $j$ does not depend on the current state $i$, i.e.,

$$T(i, j) = T(i', j) = p_j.$$ 

Let $r(i) =$ the reward you collect if you stop at state $i$.
Let $W(j) =$ the expected optimal value if you start at state $j$.

Let $C = \sum_{j=1}^{m} p_j W(j)$.

For the dead state $\Delta = 0$, $r(0) = 0$ and $W(0) = 0$.

For a nondead state $i$:

Find the expected value if you choose to continue.

Simplify the dynamic programming equation

$$W(i) = \max \{r(i), \sum_{j=0}^{m} T(i, j)W(j)\}.$$
Suppose we have a state independent problem with $m+1$ states $\{0, 1, ..., m\}$ and suppose 0 is the dead state $\Delta$.

For the dead state, the transition probabilities are $T(0, 0) = 1$, $T(0, j) = 0$ otherwise.

For all other states, the probability $p_j$ of transition to a next state $j$ does not depend on the current state $i$, i.e.,

$$T(i, j) = T(i', j) = p_j.$$  

Let $r(i)$ = the reward you collect if you stop at state $i$.
Let $W(j)$ = the expected optimal value if you start at state $j$.
Let $C = \sum_{j=1}^{m} p_j W(j)$.

For the dead state $\Delta = 0$, $r(0) = 0$ and $W(0) = 0$. For a nondead state $i$, the expected value if you continue is

$$C = \sum_{j=0}^{m} T(i, j) W(j) = \sum_{j=0}^{m} p_j W(j) = \sum_{j=1}^{m} p_j W(j).$$

$C$ does not depend on the current state $i$.

The dynamic programming equation $W(i) = \max\{r(i), \sum_j T(i, j) W(j)\}$ simplifies to
**State-independent dynamic programming equation.**

\[ W(i) = \max \{ r(i), C \} \]

- \( r(i) \) is the value you get for stopping and
- \( C = \sum_{j=1}^{m} p_j W(j) \) is the expected value for continuing.

\[ r(i) = \text{the value you get for stopping}, \]
\[ C = \sum_{j=1}^{m} W(j) = \text{expected value for continuing}. \]

If \( r(i) \geq C \) the reward is *acceptable* and

\[ W(i) = \max \{ r(i), C \} = r(i) \]

we should stop and collect it (even in the case of equality, we must stop in order to avoid an infinite loop).

If \( r(i) < C \) the reward is *unacceptable* and

\[ W(i) = \max \{ r(i), C \} = C \]

we should continue.
Suppose the states are listed in order of increasing rewards. Thus \( r(0) = 0 \leq r(1) \leq r(2) \leq r(3) \leq \ldots \leq r(m) \).

Since \( W(i) = \max\{r(i), C\} \), we also have
\[
W(0) = 0 \leq W(1) \leq W(2) \leq \ldots \leq W(m).
\]

Let \( i^* \) be the first state with an acceptable reward \( r(i^*) \).
\[
\therefore r(1) \leq r(2) \leq \ldots \leq r(i^* - 1) < C \leq r(i^*) \leq \ldots \leq r(m).
\]

For nondead states \(< i^* \), we continue and get \( W(i) = C \).

For states \( \geq i^* \), we stop and get \( W(i) = r(i) \).

Find the probability that the next state is dead or \( \geq i \).

Find the conditional probability that the next state = \( j \) given that the next state is dead or \( \geq i \).

Find the expected optimal value given that the next state is dead or \( \geq i \).
Suppose the states are listed in order of increasing rewards. Thus \( r(0) = 0 \leq r(1) \leq r(2) \leq r(3) \leq \ldots \leq r(m) \).

Since \( W(i) = \max\{r(i), C\} \), we have
\[
W(0) = 0 \leq W(1) \leq W(2) \leq \ldots \leq W(m).
\]

Let \( i^* \) be the first state with an acceptable reward \( r(i^*) \).
\[
\therefore r(1) \leq r(2) \leq \ldots \leq r(i^* - 1) < C \leq r(i^*) \leq \ldots \leq r(m).
\]

For nondead states \( < i^* \), we continue and get \( W(i) = C \).
For states \( \geq i^* \), we stop and get \( W(i) = r(i) \).

\( p_i \) = the probability the next state is \( i \).

The probability that the next state is dead or \( \geq i \):
\[
p_0 + p_i + p_{i+1} + \ldots + p_m = p_0 + \sum_{j=i}^{m} p_j.
\]

The conditional probability that the next state = \( j \) given that the next state is dead or \( \geq i \):
\[
p_j / [p_0 + \sum_{j=i}^{m} p_j].
\]

The expected optimal value given that the next state is dead or \( \geq i \):
\[
\sum_{j=i}^{m} [p_j / (p_0 + \sum_{j=i}^{m} p_j)] W(j) = [\sum_{j=i}^{m} p_j W(j)] / [p_0 + \sum_{j=i}^{m} p_j].
\]
\[ \sum_{j=i}^{m} \left[ \frac{p_j}{p_0 + \sum_{j=i}^{m} p_j} \right] W(j) = \left[ \sum_{j=i}^{m} p_j W(j) \right] / \left[ p_0 + \sum_{j=i}^{m} p_j \right]. \]

If \( i \) is acceptable, \( W(j) = r(j) \) for \( j \geq i \) and this formula, call it \( C_i \), becomes:
\[
C_i = \left[ \sum_{j=i}^{m} p_j r(j) \right] / \left[ p_0 + \sum_{j=i}^{m} p_j \right].
\]

**Lemma.** \( i^* \) is the first state with an acceptable reward iff it is the first state such that
\[
r(i^*) \geq C_{i^*} = \left[ \sum_{j=i^*}^{m} p_j r(j) \right] / \left[ p_0 + \sum_{j=i^*}^{m} p_j \right].
\]

**Proof \( \Rightarrow \).** Suppose \( i^* \) is the first state with an acceptable reward. By the above, \( W(i) = C \) for \( i < i^* \) and \( W(i) = r(i) \) for \( i \geq i^* \).

Write \( C \) in terms of \( C \) and \( r(i) \)
\[
C = \sum_{j=1}^{m} W(j) = ??
\]
\[ \sum_{j=i}^{m} \left[ \frac{p_j}{(p_0 + \sum_{j=i}^{m} p_j)} \right] W(j) = \frac{\sum_{j=i}^{m} p_j W(j)}{p_0 + \sum_{j=i}^{m} p_j}. \]

If \( i \) is acceptable, \( W(j) = r(j) \) for \( j \geq i \) and this formula, call it \( C_i \), becomes:

\[ C_i = \frac{\sum_{j=i}^{m} p_j r(j)}{p_0 + \sum_{j=i}^{m} p_j}. \]

**Lemma.** \( i^* \) is the first state with an acceptable reward iff it is the first state such that

\[ r(i^*) \geq C_{i^*} = \frac{\sum_{j=i^*}^{m} p_j r(j)}{p_0 + \sum_{j=i^*}^{m} p_j}. \]

**Proof.** \( \Rightarrow \). Suppose \( i^* \) is the first state with an acceptable reward. Then \( W(i) = C \) for \( i < i^* \) and \( W(i) = r(i) \) for \( i \geq i^* \).

\[ \therefore C = \sum_j p_j W(j) = \sum_{j=1}^{i^*-1} p_j W(j) + \sum_{j=i^*}^{m} p_j W(j) \]

\[ \therefore C = \sum_{j=1}^{i^*-1} p_j C + \sum_{j=i^*}^{m} p_j r(j) \]

\[ \therefore C - \sum_{j=1}^{i^*-1} p_j C = \sum_{j=i^*}^{m} p_j r(j) \]

\[ \therefore C(1 - \sum_{j=1}^{i^*-1} p_j) = \sum_{j=i^*}^{m} p_j r(j) \]

\[ \therefore C = \left[ \sum_{j=i^*}^{m} p_j r(j) \right] / \left[ 1 - \sum_{j=1}^{i^*-1} p_j \right] \]

\[ \therefore C = \left[ \sum_{j=i^*}^{m} p_j r(j) \right] / [p_0 - \sum_{j=i^*}^{m} p_j] = C_{i^*}. \]

\( i^* \) is the first state with an acceptable reward iff

\[ r(i^* - 1) < C \leq r(i^*) \]

iff \( r(i^* - 1) < C_{i^*} \leq r(i^*). \quad \square \]
To find \( r(i^*) \), calculate, in order \( C_1, C_2, C_3, \ldots \).

For each \( i \) check if \( C_i \leq r(i) \).

The first \( i \) for which this is true will be the first state \( i^* \) with an acceptable reward \( r(i^*) \).

In each round of a game, a die is rolled.
If the roll is a 1, the game stops and you get nothing.
Otherwise you may choose
to continue or
to stop and collect $1 for each dot that turns up.
When should you stop and collect your money?

Instead of adding a new dead state 0, note that 1 is already a dead state. To make the dead state 1 be 0, we identify a state \( i \) with the outcome of getting \( i+1 \) dots.

Number of dots: 1, 2, 3, 4, 5, 6.
The states are: \( i = 0, 1, 2, 3, 4, 5 \).
The rewards are: \( r(i) = 0, 2, 3, 4, 5, 6 \) in dollars. Thus for \( i \geq 1, r(i) = i + 1 \). The probabilities are \( p_j = 1/6 \) for all states \( j \).

For \( i = 1, 2, 3, 4, 5 \) calculate
\[
C_i = \frac{\sum_{j=0}^{m} p_j r(j)}{p_0 + \sum_{j=0}^{m} p_j}
\]
and check if \( C_i \leq r(i) \), i.e. if, \( i = i^* \) = first state with an acceptable reward.
\[ C_i = \frac{\sum_{j=i}^{m} p_j r(j)}{p_0 + \sum_{j=i}^{m} p_j} \]
\[ = \frac{\sum_{j=i}^{5} \frac{1}{6} (j + 1)}{\frac{1}{6} + \sum_{i}^{5} \frac{1}{6}} \]
\[ = \frac{\sum_{j=i}^{5} (j + 1)}{1 + \sum_{i}^{5} 1} \]
\[ = \frac{\sum_{j=i}^{5} (j + 1)}{1 + (5 - (i - 1))} \]
\[ = \frac{6 - i + \sum_{j=i}^{5} j}{7 - i} \]

\[ C_1 = \frac{6 - 1 + \sum_{j=1}^{5} j}{7 - 1} \]
\[ = \frac{5 + 1 + 2 + 3 + 4 + 5}{6} = \frac{20}{6} = 3.33 > 2 = r(1) \]

\[ C_2 = \frac{6 - 2 + \sum_{j=2}^{5} j}{7 - 2} \]
\[ = \frac{4 + 2 + 3 + 4 + 5}{5} = \frac{18}{5} = 3.6 > 3 = r(2) \]

\[ C_3 = \frac{6 - 3 + \sum_{j=3}^{5} j}{7 - 3} \]
\[ = \frac{3 + 3 + 4 + 5}{4} = \frac{15}{4} = 3.75 < 4 = r(3), \quad i^* = 3. \]

Thus the minimum acceptable reward is $4.00. \square
A Markov chain of the following form is called a random walk with reflecting barriers.

This is a rough model of a volatile stock which ranges randomly between some upper bound at which it is judged overpriced and listed as a “sell” and lower bound at which it is judged undervalued and listed as a “buy”.

A stock portfolio consists of two investments a stock and a bond.

The stock is a volatile stock which is a random walk with reflecting barriers as above with probability $p$ of a unit increase, probability $1 - p$ of decrease except when at a barrier.

The bond is a government bond which consistently earns an interest of $\beta$ and hence grows each year by a factor of $1 + \beta$. 
At any time, the investor can withdraw funds from either investment.

He can either keep the withdrawal or deposit it in the other investment.

His goal is to maximize the total amount withdrawn.

In the finite horizon case there is a fixed terminal time at which time he withdraws all funds from both investments.

In the discounted infinite horizon case, he must maximize his total withdrawals over an infinite amount of time with some discount rate $\alpha < 1$. 
The state at any time is a vector \((x, k, y, l)\) where
\[\begin{align*}
x &= \text{the current price for one share of the stock.} \\
k &= \text{the number of shares he owns of this volatile stock.} \\
y &= \text{the current price for one share of the bond.} \\
l &= \text{the number of shares he owns of the bond.}
\end{align*}\]

The \textit{value} of the stocks and bonds is ??

In the finite horizon case, the final reward is the value of the stocks and bonds at time \(T = R(x, k, y, l) = ??\)

An \textit{action} is a vector \((a, b)\) where \(a\) is the number of stocks sold (or equivalently, withdrawn) and \(b\) is the number of bond shares sold.

The reward \(r(x, k, y, l, a, b) = ?? = \text{the current withdrawal} = \text{the current sale.}\)
The state at any time is a vector \((x, k, y, l)\) where

- \(x\) = the current price for one share of the stock.
- \(k\) = the number of shares he owns of this volatile stock.
- \(y\) = the current price for one share of the bond.
- \(l\) = the number of shares he owns of the bond.

The **value** of the stocks and bonds is \(kx + ly\). In the finite horizon case, the final reward is the value \(R(x, k, y, l) = kx + ly\) of the stocks and bonds at time \(T\).

An **action** is a vector \((a, b)\) where \(a\) is the number of stocks sold (or equivalently, withdrawn) and \(b\) is the number of bond shares sold.

The reward \(r(x, k, y, l, a, b) = ax + by = \text{the current withdrawal} = \text{the current sale.}\)
Buying is the negative of selling. So $a = 3$ means three stock shares are sold and $b = -2$ means two bond shares are bought, i.e., a negative withdrawal is a buy.

A policy is a pair of functions $(u^a(x, k, y, l), u^b(x, k, y, l))$ where $u^a(x, k, y, l) =$ the number $a$ of stock shares to sell and $u^b(x, k, y, l) =$ the number of $b$ bond shares to sell. Each action depends on the current state $(x, k, y, l)$. Nonstationary policies (used in the finite horizon case) also depend on the current time $n$: $(u^a_n(x, k, y, l), u^b_n(x, k, y, l))$.

If you the amount you buy exceeds the amount you sell, then $ax + by < 0$.

We now assume this doesn’t happen.

**Assumption.** No new money is used to purchase stocks or bonds. When you buy one investment (stock or bond) the money used for the purchase is withdrawn from the other investment.

Thus the total sale $ax + by$ cannot be negative. Also, the current sale cannot exceed the current value. Thus $0 \leq ax + by \leq kx + ly$. 
Now calculate the transition probability \( T_{(a,b)}((x, k, y, l), (x^*, k^*, y^*, l^*)) \) for going from state \((x, k, y, l)\) to state \((x^*, k^*, y^*, l^*)\) if the action you take is \((a, b)\).

With probability one we must have:

\[
\begin{align*}
k^* &= k, \\
l^* &= l, \\
y^* &= y.
\end{align*}
\]

Recall:

\( x \) = the current price for one share of the stock.
\( k \) = the number of shares he owns of this volatile stock.
\( y \) = the current price for one share of the bond.
\( l \) = the number of shares he owns of the bond.
\( a \) = the number of stocks sold.
\( b \) = the number of bond shares sold.

A bond grows each year by a factor of \( 1+\beta \).
Now calculate the transition probability 
\( T_{(a,b)}((x, k, y, l), (x^*, k^*, y^*, l^*)) \) for going
from state \((x, k, y, l)\)
to state \((x^*, k^*, y^*, l^*)\) if the action you take is \((a, b)\).
With probability one we must have:
\[
k^* = k - a, \quad l^* = l - b, \quad y^* = (1 + \beta)y.
\]
If the three conditions are true, then the transition probability
\( T_{(a,b)}((x, k, y, l), (x^*, k^*, y^*, l^*)) = \) the transition probability
\( T(x, x^*) \) that the price of the volatile stock will go from \( x \) to
\( x^* \). This is the probability given by the random walk.

In the barrier cases,
\[
x = 0 \rightarrow x^* = 1 \quad \text{and} \quad x = N \rightarrow x^* = N - 1 \quad \text{have probability 1. Otherwise}
\]
\[
x^* = x + 1 \quad \text{has probability } p \quad \text{and} \quad x^* = x - 1 \quad \text{has probability } 1 - p.
\]