CME 241 Assignent 2

Justin Cramer

1/23/2023

1.

a.

$$\text{States } \mathcal{S} = \{0, \ldots, 100\}$$
$$\text{Terminal states } \mathcal{T} = \{100\}$$
$$\mathcal{N} = \mathcal{S} - \mathcal{T}$$

b.

$$\text{States with latters from them } \mathcal{L} = \{1, 4, 8, 21, 28, 50, 71, 80\}$$
$$\text{States with snakes from them } \mathcal{K} = \{32, 36, 48, 62, 88, 95, 97\}$$
$$\mathbb{P}(S_{t+1} = f(s+i) | S_t = s) = 1/6, \text{ for } 1 \leq i \leq 6 \text{ and } s \in \mathcal{N}, s \notin \mathcal{L} \cup \mathcal{K}$$
$$\text{Where } f(x) = \text{position latter/snake leads to from x if } x \in \mathcal{L} \cup \mathcal{K}$$
$$f(x) = 100 \text{ if } x > 100$$
$$f(x) = x \text{ otherwise}$$

c.

In [372…
```
from dataclasses import dataclass
from typing import Mapping, Dict, Tuple
import sys
import os
import numpy as np
import itertools
import matplotlib.pyplot as plt
sys.path.append(os.path.abspath("/Users/justincramer/Documents/Coding/CME241/RL-book/"))

from rl import markov_process
from rl.markov_process import FiniteMarkovProcess
from rl.distribution import Categorical, Constant, Categorical, FiniteDistribution
from rl.markov_process import FiniteMarkovRewardProcess, MarkovRewardProcess
```

In [405…
```
states = set(np.arange(101))
terminal = set([100])
nonterminal = states - terminal
def f(x):
    d = {1: 38, 4: 14, 8: 30, 21: 42, 28: 76, 32: 10, 36: 6, 48: 26, 50: 67, 62: 18,
         71: 92, 80: 99, 88: 24, 95: 56, 97: 78}
    return d.get(x, min(x, 100))
```

In [406…
```
@dataclass(frozen=True)
class StateSL:
    position: int
```

```
class SLFinite(FiniteMarkovProcess[StateSL]):
    def __init__(self):
        super().__init__(self.get_transition_map())
    def get_transition_map(self) -> Mapping[StateSL, FiniteDistribution[StateSL]]:
        d: Dict[StateSL, Categorical[StateSL]] = {}
        for s in list(nonterminal):
            states_probs_map: Mapping[StateSL, float] = {
                StateSL(f(s + i)): 1 / 6 for i in range(1, 7)
            }
            d[StateSL(s)] = Categorical(states_probs_map)
        return d
```

In [407…

```
lens = []
for trace in traces:
    lens.append(len(list(trace)))
```

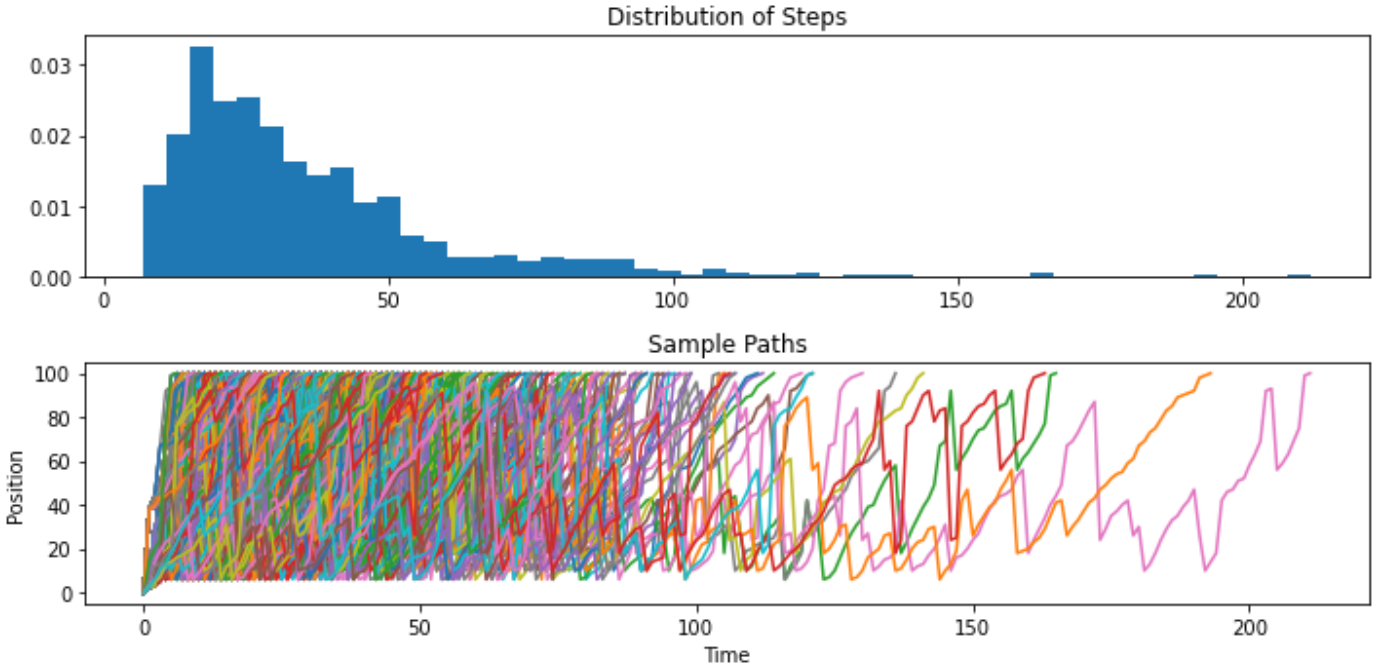
In [417…

```
sl = SLFinite()
traces = itertools.islice(sl.traces(Categorical({list(sl.transition_map.keys())[0]: 1})))
steps = []
paths = []
for trace in traces:
    paths.append(list(trace))
    steps.append(len(paths[-1]))
```

d.

In [418…

```
fig, ax = plt.subplots(2, 1, figsize=(10, 5))
ax[0].hist(steps, density=True, bins=50)
ax[0].set_title('Distribution of Steps')
for s in paths:
    s = [a.state.position for a in s]
    ax[1].plot(list(range(len(s))), s)
ax[1].set_title('Sample Paths')
ax[1].set_xlabel('Time')
ax[1].set_ylabel('Position')
fig.tight_layout()
```



e.

In [414…

```
class SLReward(FiniteMarkovRewardProcess[StateSL]):
    def __init__(self):
        super().__init__(self.get_transition_reward_map())

    def get_transition_reward_map(self) -> Mapping[StateSL, FiniteDistribution[Tuple[Sta
        d: Dict[StateSL, Categorical[Tuple[StateSL, float]]] = {}
        for state in non_terminal:
            state_probs_map: Mapping[Tuple[StateSL, float], float] = {}
            for s in list(nonterminal):
                states_probs_map = {
                    (StateSL(f(s + i)), 1): 1 / 6 for i in range(1, 7)
                }
                d[StateSL(s)] = Categorical(states_probs_map)
        return d
```

In [415…

```
sl = SLReward()
SL_MRP.display_value_function(1)
```

```
{NonTerminal(state=StateSL(position=0)): 34.026,
 NonTerminal(state=StateSL(position=1)): 34.867,
 NonTerminal(state=StateSL(position=2)): 34.598,
 NonTerminal(state=StateSL(position=3)): 34.416,
 NonTerminal(state=StateSL(position=4)): 34.58,
 NonTerminal(state=StateSL(position=5)): 34.316,
 NonTerminal(state=StateSL(position=6)): 34.051,
 NonTerminal(state=StateSL(position=7)): 33.79,
 NonTerminal(state=StateSL(position=8)): 33.632,
 NonTerminal(state=StateSL(position=9)): 33.325,
 NonTerminal(state=StateSL(position=10)): 33.018,
 NonTerminal(state=StateSL(position=11)): 32.726,
 NonTerminal(state=StateSL(position=12)): 32.462,
 NonTerminal(state=StateSL(position=13)): 32.229,
 NonTerminal(state=StateSL(position=14)): 32.031,
 NonTerminal(state=StateSL(position=15)): 31.485,
 NonTerminal(state=StateSL(position=16)): 31.173,
 NonTerminal(state=StateSL(position=17)): 30.977,
 NonTerminal(state=StateSL(position=18)): 30.874,
 NonTerminal(state=StateSL(position=19)): 30.831,
 NonTerminal(state=StateSL(position=20)): 30.843,
 NonTerminal(state=StateSL(position=21)): 31.302,
 NonTerminal(state=StateSL(position=22)): 29.302,
 NonTerminal(state=StateSL(position=23)): 29.802,
 NonTerminal(state=StateSL(position=24)): 30.257,
 NonTerminal(state=StateSL(position=25)): 30.568,
 NonTerminal(state=StateSL(position=26)): 30.918,
 NonTerminal(state=StateSL(position=27)): 30.965,
 NonTerminal(state=StateSL(position=28)): 33.214,
 NonTerminal(state=StateSL(position=29)): 32.806,
 NonTerminal(state=StateSL(position=30)): 32.984,
 NonTerminal(state=StateSL(position=31)): 32.436,
 NonTerminal(state=StateSL(position=32)): 31.724,
 NonTerminal(state=StateSL(position=33)): 31.247,
 NonTerminal(state=StateSL(position=34)): 30.792,
 NonTerminal(state=StateSL(position=35)): 30.359,
 NonTerminal(state=StateSL(position=36)): 29.386,
 NonTerminal(state=StateSL(position=37)): 29.148,
 NonTerminal(state=StateSL(position=38)): 28.745,
 NonTerminal(state=StateSL(position=39)): 28.387,
 NonTerminal(state=StateSL(position=40)): 28.062,
 NonTerminal(state=StateSL(position=41)): 27.761,
 NonTerminal(state=StateSL(position=42)): 28.212,
 NonTerminal(state=StateSL(position=43)): 27.721,
 NonTerminal(state=StateSL(position=44)): 26.326,
 NonTerminal(state=StateSL(position=45)): 26.238,
 NonTerminal(state=StateSL(position=46)): 26.115,
 NonTerminal(state=StateSL(position=47)): 25.953,
 NonTerminal(state=StateSL(position=48)): 24.891,
 NonTerminal(state=StateSL(position=49)): 24.777,
 NonTerminal(state=StateSL(position=50)): 25.961,
 NonTerminal(state=StateSL(position=51)): 25.707,
 NonTerminal(state=StateSL(position=52)): 25.378,
 NonTerminal(state=StateSL(position=53)): 24.982,
 NonTerminal(state=StateSL(position=54)): 24.547,
 NonTerminal(state=StateSL(position=55)): 24.092,
 NonTerminal(state=StateSL(position=56)): 25.061,
 NonTerminal(state=StateSL(position=57)): 24.184,
 NonTerminal(state=StateSL(position=58)): 23.403,
 NonTerminal(state=StateSL(position=59)): 22.606,
 NonTerminal(state=StateSL(position=60)): 21.933,
 NonTerminal(state=StateSL(position=61)): 21.365,
 NonTerminal(state=StateSL(position=62)): 19.162,
 NonTerminal(state=StateSL(position=63)): 18.924,
```

```
 NonTerminal(state=StateSL(position=64)): 18.715,
 NonTerminal(state=StateSL(position=65)): 17.825,
 NonTerminal(state=StateSL(position=66)): 17.896,
 NonTerminal(state=StateSL(position=67)): 17.957,
 NonTerminal(state=StateSL(position=68)): 17.655,
 NonTerminal(state=StateSL(position=69)): 17.493,
 NonTerminal(state=StateSL(position=70)): 17.466,
 NonTerminal(state=StateSL(position=71)): 18.358,
 NonTerminal(state=StateSL(position=72)): 18.326,
 NonTerminal(state=StateSL(position=73)): 18.32,
 NonTerminal(state=StateSL(position=74)): 15.846,
 NonTerminal(state=StateSL(position=75)): 16.523,
 NonTerminal(state=StateSL(position=76)): 17.3,
 NonTerminal(state=StateSL(position=77)): 17.831,
 NonTerminal(state=StateSL(position=78)): 18.137,
 NonTerminal(state=StateSL(position=79)): 18.285,
 NonTerminal(state=StateSL(position=80)): 21.154,
 NonTerminal(state=StateSL(position=81)): 20.584,
 NonTerminal(state=StateSL(position=82)): 21.966,
 NonTerminal(state=StateSL(position=83)): 21.013,
 NonTerminal(state=StateSL(position=84)): 19.973,
 NonTerminal(state=StateSL(position=85)): 19.172,
 NonTerminal(state=StateSL(position=86)): 18.216,
 NonTerminal(state=StateSL(position=87)): 17.162,
 NonTerminal(state=StateSL(position=88)): 13.668,
 NonTerminal(state=StateSL(position=89)): 15.296,
 NonTerminal(state=StateSL(position=90)): 13.737,
 NonTerminal(state=StateSL(position=91)): 14.365,
 NonTerminal(state=StateSL(position=92)): 12.48,
 NonTerminal(state=StateSL(position=93)): 10.84,
 NonTerminal(state=StateSL(position=94)): 9.291,
 NonTerminal(state=StateSL(position=95)): 5.115,
 NonTerminal(state=StateSL(position=96)): 4.384,
 NonTerminal(state=StateSL(position=97)): 1.361,
 NonTerminal(state=StateSL(position=98)): 1.167,
 NonTerminal(state=StateSL(position=99)): 1.0}
```

34 rolls are expected.

2. a) states are number of steps until end with $n=9$, $S=\{0,\dots,10\}$

$S_t=10$ is start, $S_t=0$ is end (terminal state)

$P(S_{t+1}=x-i \mid S_t=x)=\frac{1}{x}$ for $1\le i\le x$, $1\le x\le 10$

b) $\sum_{i=1}^{10}\frac{1}{i}=2.93$

c) expected number when 1 step away $E[1]=1$

2 steps away $E[2]=\frac{1}{2}(E[1]+1)+\frac{1}{2}\cdot 1$

3 steps away $E[3]=\frac{1}{3}(E[2]+1)+\frac{1}{3}(E[3]+1)+\frac{1}{3}$

N steps away $E[N]=\frac{1}{N}\sum_{i=1}^{N-1}E[N-i]+1$

$E[N-1]=\frac{1}{N-1}\sum_{i=1}^{N-2}E[N-1-i]+1$

$NE(N)-(N-1)E(N-1)=E(N-1)+1$

$E(N)=E(N-1)+\frac{1}{n}$

$E(N-1)=E(N-2)+\frac{1}{n-1}$

$\vdots$

$E(N)=\sum_{i=1}^{N}\frac{1}{i}$

For n hiking steps, $E(n+1)=\sum_{i=1}^{n+1}\frac{1}{i}$

3)

a) $V^*(s) = \max_{a \in [0,1]} \{R(s,a) + 0.5 \sum_{s' \in N} P(s,a,s') V^*(s')\}$

$R(s,a) = E[r_{t+1} | s_t = s] = (1-a)a + (1+a)(1-a)$

$= 1 + a - 2a^2$

$P(s,a,s+1) = a \quad P(s,a,s) = 1-a$

$V^*(s) = \max_{a \in [0,1]} \{1 + a - 2a^2 + 0.5(aV^*(s+1) + (1-a)V^*(s))\}$

$V^*(s) = V^*(s+1)$

so $V^*(s) = \max_{a \in [0,1]} \{1 + a - 2a^2 + 0.5V^*(s)\}$

b) $\frac{\partial}{\partial a}(1 + a - 2a^2 + 0.5V^*(s)) = 1 - 4a$

V^* decreasing w/ a, so choose $a = 0$

$\Pi^*(s) = 0$