1. (Travelling Salesman Problem) The Travelling Salesman Problem describes a salesman who must travel between N cities. The order in which he does so is something he does not care about, as long as he visits each once during his trip, and finishes where he was at first. Each city is connected to other close by cities. Each of those links between the cities has the cost attached. The cost describes how "difficult" it is to traverse this edge on the graph, and may be given, for example, by the cost of an airplane ticket or train ticket, or perhaps by the length of the edge, or time required to complete the traversal. The salesman wants find a least cost trip route that visits each of each of the cities exactly once and returns to the starting city.

For example, when there are four cities and the cost on the link between each pair of cities is given, the problem situation can be represented as a graph below.

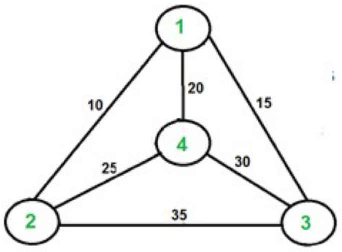


Figure 1: An example graph for the network among cities

In addition, the above graph can be expressed as an adjacency matrix as below, too.

	1	2	3	4
1	0	10	15	20
2	10	0	35	25
3	15	35	0	30
4	20	25	30	0

Figure 2: An example adjacency matrix

(a) State a DP algorithm to find the optimal trip.

- 본 문제의 Notations는 다음과 같음.
  - Decision : t $(t = 1, 2, \dots N)$
  - State : S_t
  - Action : a_t
- $V_t(S_t) = \min_{a} \left[g_t(S_t, a_t) + V_{t+1}(f_t(S_t, a_t)) \right]$
- $V_N(S_N) = 0$ $(S_0 = S_N)$

(b) Build a Python code to implement your DP algorithm. The code must be able to solve general situation (not specified for above example). That is, when an arbitrary adjacency matrix and starting city will be given, your code needs to find the optimal path. At the beginning, please use the "adj_matrix.csv" file which contains above example.

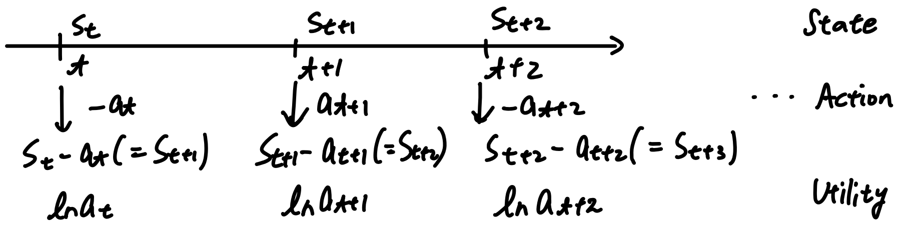
(c) Please test whether your code can be applicable for the bi-directed graph (that is, the travel cost from city 1 to city 2 can be different with the travel cost from city 2 to city 1).

- 코드 참고

2. Assume that there is a cake whose size at time t is denoted by s_t and a person who wants to eat in T periods. The initial size of the cake s_0(a positive constant) and the person needs to eat all, $s_T = 0$. The person has a psychological discount factor $0 < \frac{1}{1+r} < 1$ and an utility function $ln(a_t)$ from eating the cake as a_t at time t.

(a) Describe the maximum possible utility that the person can get from remaining cake s_t at time t. Formulate a DP to find the optimal flow of cake eating $\{a_t^* | t \in 0, 1, ..., T\}$

· 위의 description을 그림으로 나타내면 다음과 같음.



· Notations

· State: s_t $(t = 0, 1, ..., T)$

· Action(eating) : a_t $(t = 0, 1, ..., T)$

· Present value of Utility: $\frac{\ln a_t}{(1+r)^t}$ $(t = 0, 1, ..., T)$

$(= g_t(s_t, a_t))$

· 현재 시점 t 기준으로 가치함수 변환

· $$V_t(s_t) = g_t(s_t, a_t) + \sum_{k=t+1}^{T-1} \lambda^{k-t} g_{t+1}(s_{t+1}, a_{t+1}) + g_N(s_N)$$

$$= \ln a_t + \frac{1}{(1+r)^{k-t}} \sum_{k=t+1}^{T-1} \ln a_{t+1} + 0$$

· $$\tilde{V}_t(s_t) = \max_{a_t} \left\{ \ln a_t + \sum_{k=t+1}^{T-1} \frac{\ln a_{t+1}}{(1+r)^{k-t}} \right\}$$

$$= \max_{a_t} \left\{ \ln a_t + \frac{1}{1+r} \sum_{k=t+1}^{T-1} \frac{\ln a_{t+1}}{(1+r)^{k-(t+1)}} \right\}$$

$$= \max_{a_t} \left\{ \ln a_t + \frac{1}{1+r} \tilde{V}_{t+1}(s_{t+1}) \right\}$$

· $$\tilde{V}_T(s_T) = 0$$

(b) Now, let's consider the continuous-time dynamic system of this cake eating problem. The flow of cake eating and the size of the cake can be represented as $\{a(t)|t \in [0,T]\}$ and the size of the cake $\{s(t)|t \in [0,T]\}$. Describe the maximum possible utility that the person can get from initial cake $s(0)$ at time 0. Find the HJB equation to find the optimal flow of cake eating. (Hint: The corresponding continuous-time discount factor can be represented as e^{-rt}.)

- discrete → continuous 로 변환해서 접근.
- Notations
  - State: S_t $(t \in [0,T])$
  - present value of Utility : $e^{-rt} \cdot \ln a_t$
- $s(0)$이 주어졌을 때 discount factor를 고려한 Utility function은 다음과 같음.
  - $e^{-rT} \cdot h(s(T)) + \int_0^T e^{-rt} \cdot \ln a_t \, dt$
    $= e^{-rT} h(0) + \int_0^T \ln a_t \cdot e^{-rt} \, dt = \int_0^T e^{-rt} \cdot \ln a_t \cdot dt$
- 본 문제의 DP는 $\int_0^T e^{-rt} \cdot \ln a_t$를 maximize 하는게 목표

- $T = k\delta$ 로 고려하였을 때 discrete $\tilde{V}_0(s_0)$는 다음과 같음.
  - $\tilde{V}_0(s_0) = \max_{a_t} \left\{ 0 \cdot e^{-rT} + \sum_{i=0}^{k-1} \ln a_t \cdot e^{-ri} \cdot \delta \right\}$

$$= \max_{a_t} \left\{ \ln a_0 \delta + \sum_{i=1}^{k-1} \ln a_t \cdot e^{-ri} \cdot \delta \right\}$$

$$= \max_{a_t} \left\{ \ln a_0 \cdot \delta + \tilde{V}_1(s_1) \right\}$$

$$= \max_{a_t} \left\{ \ln a_0 \cdot \delta + \tilde{V}_1(s_0 - a_0 \cdot \delta) \right\},$$

  - $\tilde{V}_N(S_N) = 0$
- 위의 DP 공식을 $t = k\delta$에 대해 일반화하면 다음과 같음.
  - $\tilde{V}_k(s_k) = \max_{a_k} \left\{ e^{-rt} \ln a_k \delta + \tilde{V}_{k+1}(s_k - a_k \cdot \delta) \right\}, \quad \tilde{V}_k(S_N) = 0$
  - $\tilde{V}_k(s) = \max_{a \in A} \left[g(s,a)\delta + \tilde{V}_k(s) + \nabla_t \tilde{V}_k(s)\delta + \nabla_s \tilde{V}_k(s)' \underline{f(s,a)\delta} + o(\delta) \right]$

  이므로
  - $\tilde{V}_k(s_k) = \max_{a \in A} \left[e^{-rt} \ln a_k \cdot \delta + \tilde{V}_k(s_k) + \nabla_t \cdot \tilde{V}_k(s)\delta - \nabla_s \tilde{V}_k(s)' a_k \delta + o(\delta) \right]$

· $\lim_{k\to\infty,\, \delta\to 0} \tilde{V}_k^*(s) = V_t^*(s)$ 이므로

$$0 = \max_{a\in A}\left[e^{-rt}\cdot \ln a + \nabla_t \cdot V_t(s) - \nabla_s V_t(s)'\cdot a_k\right]$$

(c) Assume that a value function defined as

$$V_t(s) = e^{-rt}(x + y\, ln(s))$$

where x and y are constants, can represent an optimal flow $a^*(t) = rs(t)$ in problem (b). Specify x and y.

· 시점 t 에서 state가 s로 정의될 때 가치함수는 다음과 같음.

$$V_t(s) = e^{-rt}(x + y\cdot \ln(s))$$

· t와 s에 대해서 위의 식을 미분하면 다음과 같음.

$$\nabla_t \cdot V_t(s) = -r\cdot e^{-rt}(x + y\cdot \ln(s))$$

$$\nabla_s \cdot V_t(s) = \frac{1}{s}\cdot e^{-rt}\cdot y$$

· (b) 식 HJB 방정식에 대입하면 다음과 같음

$$0 = \max_{a\in A}\left[e^{-rt}\cdot \ln a + \nabla_t \cdot V_t(s) - \nabla_s V_t(s)'\cdot a_k\right]$$

$$= \max_{a\in A}\left[e^{-rt}\cdot \ln a - re^{-rt}(x+y\cdot\ln(s)) - \frac{1}{s}\cdot e^{-rt}\cdot y\cdot a\right]$$

· $a^* = r\, s(t)$ 이므로

$$\cancel{e^{-rt}}\cdot \ln r\cdot s(t) - r\cdot\cancel{e^{-rt}}(x+y\cdot\ln(s)) - \frac{1}{s}\cdot \cancel{e^{-rt}}\cdot y\cdot r\cdot s(t)]$$

$$= 0.$$

$$\ln r\cdot s - rx - ry\cdot \ln s - \frac{y}{\cancel{s}}\cdot r\cdot \cancel{s} = 0$$

$$\ln r + \ln s - rx - ry\cdot\ln s - ry = 0.$$

$$\ln r - rx + \ln s\,(1 - ry) - ry = 0.$$

$$\ln r - rx - ry + \ln s\,(1 - ry) = 0.$$

$$y = \frac{1}{r}, \quad rx = \ln r - 1 \quad \therefore\ x = \frac{\ln r - 1}{r}$$

$$y = \frac{1}{r} \quad ∎$$

3. Suppose that an investor owns a asset, the value of which fluctuates over time and which must be sold by some final time N. Let X_t denote the price at time t, and assume that $X = X_1, X_2, ... : t \geq 0$ is Markov process with values in the set $S' = [0, \infty)$. If the investor sells the asset, then he can earn a money s_t where $s_t = X_t$ is the price of the asset at time t and invest the money at a fixed rate of interest $r > 0$. Otherwise, he waits until the next period. If the asset is still held at time N, then it must be sold at a price of s where $s = X_N$ is the final value. When the investor still holds the asset at time t, a policy for selling or not selling the asset defined as

selling the asset if $s_t \geq B_t$

not selling the asset if $s_t < B_t$

can be a candidate to maximize the revenue of the investor at the Nth period. Find an appropriate structure of B_t to make the policy optimal.

- 본 문제에서의 Notations는 다음과 같음
  - Decision epoch: $T = \{1, \ldots, N\}$
  - State space: $S = S' \cup \{\Delta\}$, Δ is absorbing state
  - Action space: $\begin{cases} A_s = \{G, W\} & \text{if } s \in S' \ (G: \text{sell (Go)}, W: \text{wait}) \\ A_s = \{G\} & \text{if } s \in \Delta \end{cases}$
  - Rewards

$$g_t(s, a) = \begin{cases} 0 & \text{if } s \in S, A = G \\ (1+r)^{N-t} \cdot s & \text{if } s \in S', A = W \\ 0 & \text{if } s = \Delta \end{cases}$$

$$g_N(s_N) = \begin{cases} s_N & \text{if } s \in S' \\ 0 & \text{if } s \in \Delta \end{cases}$$

- Optimality 원리에 의해, 모든 t에 대해서 optimal 해야함.
  - $u_t^\pi(s_t) = g_t(s_t, a_t) + E_{s_t}^\pi\left[u_{t+1}^\pi(s_t, a_t, s_{t+1})\right] \quad (t = 1, 2, \ldots, N-1)$
  - $u_N^\pi(s_N) = g_N(s_N)$
- Case I : t=N인 경우
  - 투자자가 받는 돈은 정해져 있음.
  - 따라서, t=N 일 때는 어떤 전략도 취할 수 없기에 optimal함.
  - $u_N^\pi(s_N) = \begin{cases} s_N & \text{if } s_N \neq 0 \\ 0 & \text{if } s_N = 0. \end{cases}$

- Case II : $t = N-1$ 일 경우
  - 주어진 정책 π 를 실행할 때,
    - 만약 $S_{N-1} < B_{N-1}$: holding should be optimal on all $S_{N-1} \in [0, B_{N-1})$
      - $E_N^{\pi}[u_N^{\pi}(S_N)] = S_N \geq (1+r) S_{N-1}$
    - 만약 $S_{N-1} > B_{N-1}$: holding should be optimal on all $S_{N-1} \in [B_{N-1}, \infty)$
      - $E_N^{\pi}(u_N^{\pi}(S_N)) = S_N \leq (1+r) S_{N-1}$
    - LHS $\Leftrightarrow$ RHS 균형은 $S_{N-1} = B_{N-1}$ 에서 이루어지므로,
      $B_{N-1} = \frac{E_N^{\pi}(u_N^{\pi}(S_N))}{1+r} = \frac{S_N}{1+r}$ 일 때 optimal 하다 할 수 있음.
    - $u_{N-1}^{\pi}(S_{N-1})$은 다음과 같이 적용 가능.
      - $$u_{N-1}^{\pi}(S_{N-1}) = g_{N-1}(S_{N-1}, a_{N-1}) + E_{S_{N-1}}^{\pi}[u_N^{\pi}(S_{N-1}, a_{N-1}, S_N)]$$
        $$= \begin{cases} S_N & \text{if } S_{N-1} \neq \Delta, a_{N-1} = G \\ (1+r) S_{N-1} & \text{if } S_{N-1} \neq \Delta, a_{n-1} = W \\ 0 & \text{if } S_{N-1} = \Delta \end{cases}$$

- Case III : $t = N-2$ 일 경우
  - 주어진 정책 π에 대해서
    - 만약 $S_{N-2} < B_{N-2}$: optimal on all $S_{N-2} \in [0, B_{N-2})$
      - $E_{N-1}^{\pi}[u_{N-1}^{\pi}(S_{N-1})] \geq (1+r)^2 S_{N-2}$
    - 만약 $S_{N-2} \geq B_{N-2}$: 〃 $S_{N-2} \in [B_{N-2}, \infty)$
      - $E_{N-1}^{\pi}[u_{N-1}^{\pi}(S_{N-1})] \leq (1+r)^2 S_{N-2}$
  - 따라서, S_{N-1}과 S_N이 주어졌을 때, $S_{N-2} = B_{N-2}$ 일 때 LHS와 RHS가 균형을 가짐
  - 따라서 $B_{N-2} = \frac{E_{N-1}^{\pi}[u_{N-1}^{\pi}(S_{N-1})]}{(1+r)^2}$ 일 때, π가 optimal 하다 할수 있음.

- Case IV : 일반화 버전 ($t = N-k$ 일 때)
  - $S_{N-k} = B_{N-k}$ 일때 균형.
  - $B_{N-k} = \frac{E^{\pi}_{N-k+1}[u^{\pi}_{N-k+1}(S_{N-k+1})]}{(1+r)^{k}}$
- 따라서 $B_t = \frac{E^{\pi}_{t+1}[u^{\pi}_{t+1}(S_{t+1})]}{(1+r)^{N-t}}$ 일 때가 optimal 함 $\blacksquare$

4. (Bandit Model) Suppose that a gambler in a casino can either pay c units to pull the lever on a slot machine and that the machine pays 1 unit with the probability q and 0 units with the probability $1-q$, or decide not to play. Unfortunately, the values of the probabilities q are unknown, but the gambler gains information concerning the distribution of q each time that she chooses to play the game. The prior distribution is denoted as $f(q)$, a density with support on [0,1]. By playing the game several times, the gambler acquires the information about the distribution of q and revises her assessed probability density accordingly. Here, the gambler seeks to maximize her expected winnings. (a) Formulate this problem as a MDP. (b) Now, imagine that the casino has K slot machines and the gambler choose one of the machines when she decide to play. $i^t h$ machine plays 1 unit with the probability q_i and 0 units with the probability $1-q_i$, and the prior distribution for q_i is denoted as $f_i(q_i)$, a density with support on [0,1]. Here, she faces a tradeoff between exploiting the machine that appears to be the best based on the information collected thus far and exploring other machines that might have higher probabilities of winning. Formulate this revised problem as a MDP. (Hint: Let the decision maker observe the state of $K+1$ reward processes, where $i = 1, ..., K$ represents the option for "playing" i^{th} slot machine and the last process $K+1$ correspond to the "do not play" option.)

(a) MDP로 구현

- 본 문제에서 Notations는 다음과 같음.
  - Decision epoch: $T = \{1, 2, 3, \cdots\}$
  - State : $S_t = \{ f(q) \mid q \in [0,1] \}$
  - Action: $A_s = \{P, N\}$ (P=pull, N=not to play)
  - Reward: t 시점에서 $g_t(S_t, a_t, S_{t+1}) = g_t(f_t(q), a_t, f_{t+1}(q))$

$$g_t = \begin{cases} -c+1 & \text{if } A_s \in \{P\} \text{ \& 성공}, \ p = \int_0^1 q \cdot f_t(q)\,dq \\ -c & \text{if } A_s \in \{P\} \text{ \& 실패}, \ p = 1 - \int_0^1 q \cdot f_t(q)\,dq \\ 0. & \text{if } A_s \in \{N\}. \quad p = 1 \end{cases}$$

$$E[g_t] = \begin{cases} (1-c)\int_0^1 q \cdot f_t(q)\,dq + (-c)\cdot\left(1 - \int_0^1 q \cdot f_t(q)\,dq\right) & \text{if } A_s \in \{P\} \\ 0. & \text{if } A_s \in \{N\} \end{cases}$$

(b) · k개의 slot machines, i번째 기계 → $g_i \to 1$, $1-g_i \to 0$

· g_i의 이전 확률은 $f(g_i)$, Play vs Do not play. MDP로 구현하라.

· 본 문제 Notations은 다음과 같음.

· $T = \{0, 1, 2, \dots, N\}$.

· State: $S = \{ f_i(q_i),\ i = 1, 2, \dots, k \}$.

· Action: $A_s \in \{ P_i\ (i = 1, 2, \dots, k), N \}$.

· Rewards : $g_t(s_t, a_t) = g_t\left(f_{it}(q_i), a_{it}, f_{(i+1)t}(g_{i+1})\right)$

$$= \begin{cases} -C+1 & A_s \in \{P_i\},\ P = \int_0^1 g f(g)\, dg \\ -C & A_s \in \{P_i\},\ P = 1 - \int_0^1 g f(g)\, dg \\ 0 & A_s \in \{N\},\ P = 1 \end{cases}$$

· Expected reward $E[g_t]$

$$= \begin{cases} (-C+1) \cdot \int_0^1 g f(g)\, dg - C \cdot (1 - \int_0^1 g \cdot f(g)\, dg) \\ 0. \end{cases}$$