

# 물리학II 주제탐구 - 직접 만든 시뮬레이터로 엔트로피 증가 법칙 체험하기

20806 범준환

물리학 I 교과서에 나오는 열역학 제2법칙, 엔트로피 증가 법칙이라고도 하는 이 원리는, *어떤 고립계의 엔트로피가 열적 평형 상태에 있지 않다면, 그 계의 엔트로피는 증가해야 한다*는 원리다. 프랑스의 사디 카르노가 '카르노 기관'을 제안하고, 카르노 기관을 켈빈 경(절대온도의 '켈빈')과 루돌프 클라우지우스(독일, 열역학을 정립)가 정립한 개념이다. 현실의 여러 운동들이 비가역성을 띄는 이유기도 하다. 비가역 현상이란, 한쪽 방향으로만 일어나며 반대 방향으로서는 이뤄지지 않는 일을 말한다. 간단히 생각하면, 현실의 운동은 '확률이 높은 방향으로' 일어난다. 병 속에 다양한 색의 구슬들로 층을 쌓은 후 뚜껑을 닫고 충분히 섞으면, 층으로 나뉘어있던 구슬들이 색에 상관 없이 뒤섞이게 된다. 원래 상태인 층마다 나뉜 상태로 돌아갈 확률은 극히 낮아 일어나지 않는다.

독서감상문을 쓴 *시간의 본질을 찾아가는 물리여행*(마쓰우라 소 저) 중 눈에 띄는 내용이 있었다. '시간은 비가역성 원리와 연관 있다'에 대한 예시로 등장한 **방에 짐 놓기** 활동이다. 책의 상황을 그대로 가져와본다.

"100 × 100의 격자로 나뉜 정사각형 크기의 방이 있다고 하자. 짐 10개를 격자 한 칸에 짐 하나씩 놓는다. 이 짐들이 상하좌우로 무작위로 한 칸씩 움직일 때, 이 과정을 여러 번 반복하면 원래 짐이 배치된 대로 돌아가는 일은 일어나지 않는다"

위에서 말한 **방에 짐 놓기** 활동은 위의 열역학 제2법칙을 따른다. 병을 흔들어 구슬이 섞이는 것처럼, 초기 상태에서 무작위로 짐의 운동 방향이 결정된다. 원래 짐들이 놓인 위치로 돌아갈 확률은 극히 낮다. 특정 10개 격자에 짐이 놓여있을 경우의 수는 (짐은 전부 똑같다고 가정한다) 1이다. 100 × 100 격자에 10개의 짐을 놓는 경우의 수는  ${}_{100}C_{10} = 17310309456440$ , 즉,  $5.7769042e-14\%$ 라는 너무나도 작은 확률이 나오는 것이다. 글자들만 보니 다가오지 않으므로, 실제 시뮬레이터를 만들어 눈으로 확인해보자. 시뮬레이터는 프로그래밍 언어 Python과 라이브러리 Numpy를 이용해 구현한다. 먼저 100x100 방을 선언한다.

```
room = np.zeros((100, 100))
```

(np는 Numpy를 칭한다) 100x100의 원소가 전부 0으로 이루어진 배열을 선언했다. 그 다음으로는 짐 클래스를 선언하자.

|   |   |  |
|---|---|--|
| <pre>class Backpack():     def __init__(self, x, y):         self.x = x         self.y = y          room[x][y] = 1      def location(self):         return (self.x, self.y)</pre> | <pre>def move(self, x, y):     room[self.x][self.y] = 0     new_x = self.x + x     new_y = self.y + y     if new_x &lt; 0 or new_x &gt; 99:         new_x = self.x - x     if new_y &lt; 0 or new_y &gt; 99:         new_y = self.y - y      self.x = new_x     self.y = new_y     room[self.x][self.y] = 1</pre> | <pre>def move_random_one(self):     a = random.randint(1, 4)     if a == 1:         self.move(0, 1)     elif a == 2:         self.move(1, 0)     elif a == 3:         self.move(0, -1)     elif a == 4:         self.move(-1, 0)</pre> |
|---|---|--|

클래스 짐(Backpack)과, 메소드 location, move, move\_random\_one이다. 짐은 필드로 x와 y 좌표를 가진다. 초기화 시 room의 해당 좌표의 원소 값을 1로 바꿈으로써 짐이 그 위치에 있음을 알린다. 메소드 location으로 현재 좌표를 알 수 있다. 메소드 move는 인자 x, y만큼 짐을 움직인다. move\_random\_one은 상하좌우 중 한 방향을 무작위로 고른 다음 한 칸 이동시킨다.

방에 짐 10개를 두자. 코드의 간결함을 위해 (0,0), (9,9) ... (99, 99)에 짐을 두자.

```
backpacks = list()
for i in range(0, 100, 9):
    backpacks.append(Backpack(i, i))
```

그 다음 방의 상태를 출력하는 함수를 정의한다. 짐이 없으면 공백, 짐이 있으면 0로 출력한다.

```
def print_room():
    for x in room:
        print(end="|")
        for y in x:
            if y == 0:
                print(" ", end="|")
            else:
                print("0", end="|")
        print()
    print()
```

위 함수로 출력한 초기 상태의 방의 모습은 다음과 같다.

[초기 모습]

이 모습은 방에 짐을 놓는 17310309456440가지 중 한 가지의 모습이다. 짐을 무작위로 이동시키기 전에, 예측을 해보자. 짐을 무작위로 움직이는 횟수가 많아질수록 짐이 일렬로 있었다는 정보는 희미해질 것이다. 짐을 500번 옮겨 100번대의 방의 상태를 비교하자.

```
for n in range(1,501):
    for b in backpacks:
        b.move_random_one()

    if n%100==0:
        print(f"[{n}번째 시행]\n")
        print_room()

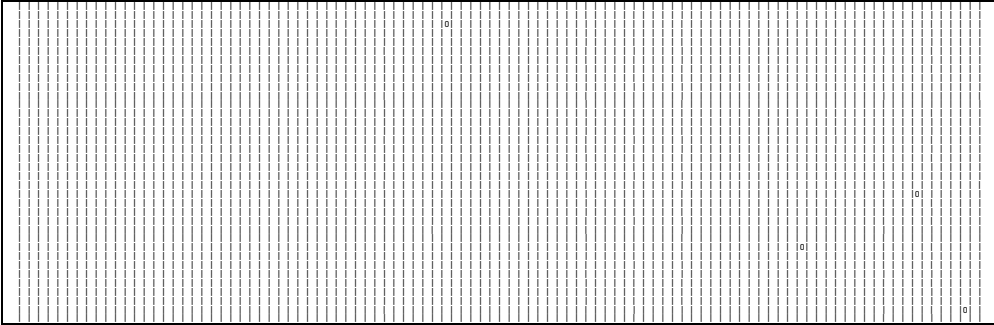
        for b in backpacks:
            print(b.location())

        print("\n\n\n\n\n\n\n")
        time.sleep(0.1)
```

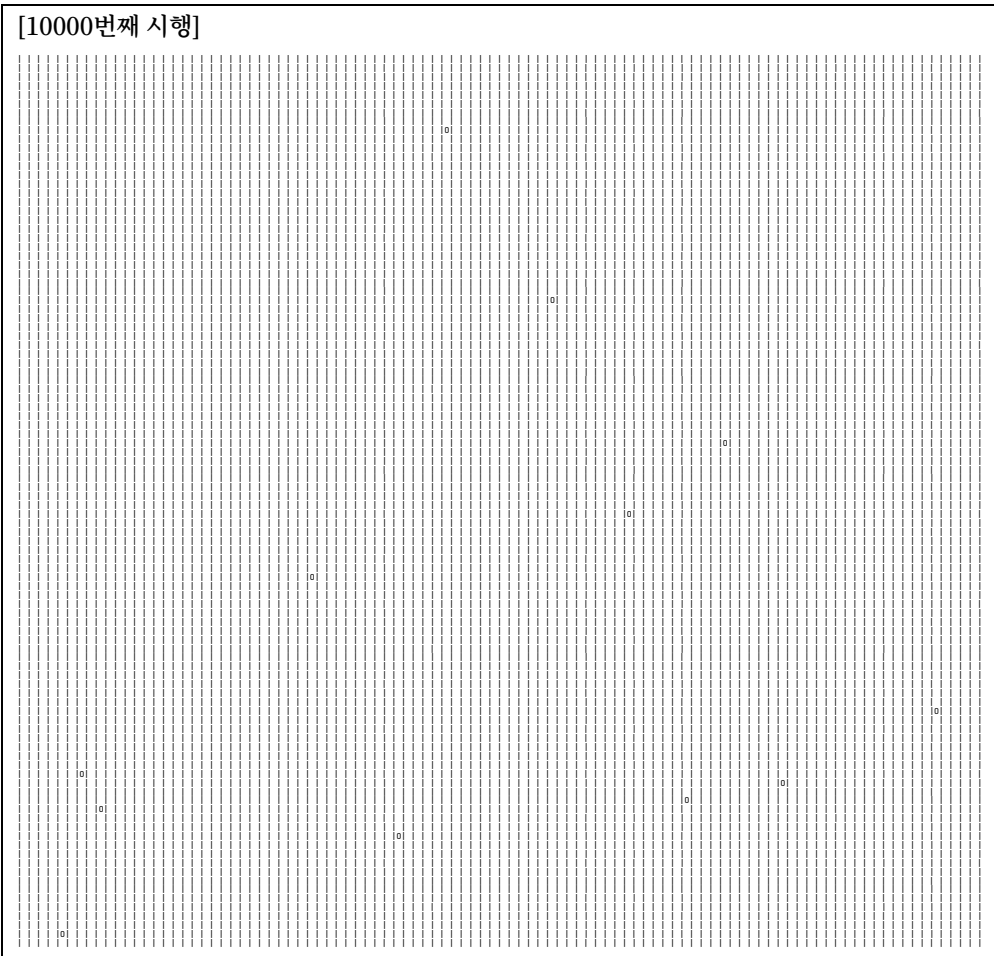
| [100번째 시행] | 좌표   |
|------------|--|
|            | (9, 5)<br>(17, 3)<br>(7, 17)<br>(23, 41)<br>(27, 25)<br>(39, 41)<br>(49, 57)<br>(60, 54)<br>(72, 78)<br>(83, 77)<br>(95, 89)<br>(91, 95) |

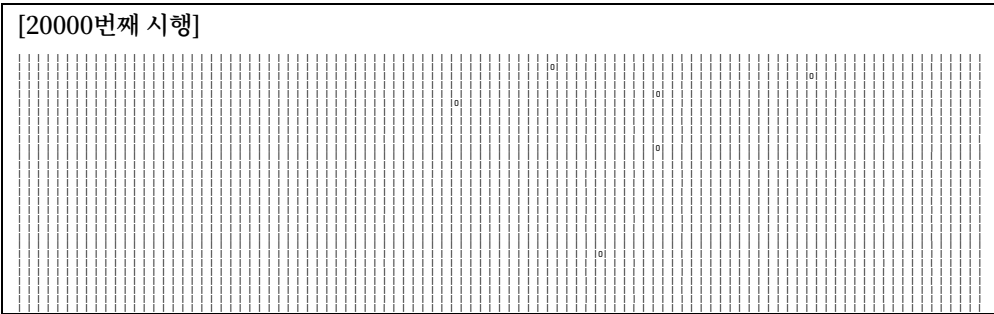




|  |          |
|--|----------|
|  | (98, 98) |
|--|----------|

점들이 원래 배치로부터 많이 벗어났지만, 아직 일렬로 있었다는 추측을 못할 정도로 흩어지진 않았다. 시행의 수가 부족한 것으로 여기고, 시행의 수를 20000번으로 늘려본다.

|   |          |
|---|----------|
| [10000번째 시행]  | 좌표       |
|  | (43, 73) |
|   | (58, 30) |
|   | (8, 44)  |
|   | (80, 6)  |
|   | (27, 55) |
|   | (81, 79) |
|   | (98, 4)  |
|   | (73, 95) |
|   | (87, 39) |
|   | (51, 63) |
|   | (84, 8)  |
|   | (83, 69) |

|  |          |
|--|----------|
| [20000번째 시행]   | 좌표       |
|  | (1, 55)  |
|  | (48, 82) |
|  | (22, 60) |
|  | (49, 27) |

|  |          |
|--|----------|
|  | (2, 82)  |
|  | (91, 87) |
|  | (4, 66)  |
|  | (37, 91) |
|  | (66, 4)  |
|  | (10, 66) |
|  | (49, 45) |
|  | (5, 45)  |

10000번째부터 일렬인 정보가 희미하고, 20000번째는 초기에 일렬이었다는 사실을 추측하기도 어렵다. 시간과 지면의 양이 부족해 더 시행하기는 어려우니 위의 시행 결과들로만 결론을 내리자면, 짐을 옮기는 행동을 시행하는 횟수가 많아질수록 일렬로 있었다는 정보는 희미해지고 다시 일렬로 돌아갈 확률은 지극히 낮으므로, 확률이 높은 쪽으로 흘러가는 자연의 법칙에 따라 짐은 흩어지게 된다. 이렇게 엔트로피 증가 법칙과 비가역성 운동을 눈으로 확인해볼 수 있었다.

## 참고 문헌

- 물리학 I (2015), 비상교육
- 시간의 본질을 찾아가는 물리여행(마쓰우시 소)
- 열역학 서론 1 <http://contents.kocw.or.kr/KOCW/document/2016/chungnam/leewanghee/1.pdf>
- 열역학 제2법칙의 역사 <https://amiooya.com/53>

ps) 1000000번 시행하면 어떻게 될까?

| [1000000번째 시행] | 좌표       |
|----------------|----------|
|                | (82, 24) |
|                | (4, 52)  |
|                | (17, 95) |
|                | (36, 60) |
|                | (0, 16)  |
|                | (46, 36) |
|                | (87, 7)  |
|                | (64, 14) |
|                | (41, 27) |
|                | (70, 50) |
|                | (30, 88) |
|                | (54, 86) |