레포트 2: 연습문제 풀이 (1주)

- 작성자 – SW융합대학 컴퓨터소프트웨어공학과 20223519 전한결
- 과목 – 2023학년도 1학기 이산수학(12744) Report 2
- 과제 제시일 – 2022년 3월 28일
- 과제 마감일시 – 2022년 4월 3일 24:00

#9

에라토스테네스의 체를 이용하여 100보다 작은 모든 소수를 구하라.

풀이 1

에라토스테네스의 체를 활용하여 100보다 작은 모든 소수를 구하는 방법 중 하나는, 2 이상 100 이하의 모든 수를 오름차순으로 써놓은 다음, 표에 있는 수 중 소수로 결정되지 않은 수들 중에서 가장 작은 수를 소수로 결정하고 그 수를 제외한 그 수의 배수를 모두 제거하는 세션을 더 이상 할 수 없을 때까지 반복하여 소수를 알아내는 방법이다.

이 방법으로 소수를 구해보도록 하겠다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

위 그림은 2부터 100까지의 모든 자연수를 나열해놓은 것이다. 이 수들 중에서는 아직 소수로 결정된 수가 없으므로 2가 남아있는 수들 중 가장 작은 수이다. 따라서 2를 소수로 결정하고 2를 제외한 2의 배수를 모두 목록에서 제거한다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

이제 2는 소수로 결정되었으므로 3이 가장 작은 수가 된다. 따라서 3을 소수로 결정하고 3을 제외한 3의 배수들을 목록에서 제거한다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

이제 2와 3은 소수로 결정되었으므로 5가 가장 작은 수가 된다. 따라서 5를 소수로 설정하고 5를 제외한 5의 배수들을 목록에서 제거한다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

이와 같은 세션을 목록의 모든 수가 소수로 결정될 때까지 반복한다.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

그렇게 하면 목록에 남아있는 수는 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97인 것을 알 수 있다.

이 범위에서의 에라토스테네스 세션의 경우 11부터는 자신을 제외한 배수가 목록에 없음이 보장되어있는데, 이는 확인하려는 수를 n이라고 했을 때 $n(n-1)$이 목록의 상한을 넘기 때문이다.

n을 소수로 결정한 세션에서 목록에서 삭제하는 수는 $kn\ (k \geq 2)$으로 표현되는 수인데, 이미 $k < n$인 k들에 대한 배수들은 이전 세션에서 지워졌을 것이기 때문에, 실제로는 n^2 이상인 수들에 대해서 배수를 지우는 과정을 진행하는 것으로 소수 목록을 알아낼 수 있다.

풀이 1의 한계

컴퓨터 알고리즘으로 풀이 1을 구현할 때에는 2부터 상한까지의 모든 수를 담은 리스트를 사전에 구비해야 하므로 많은 메모리를 필요로 한다.

18세기 말 가우스와 르장드르는 특정한 수 이하의 자연수 중 소수의 개수를 반환하는 소수 계량 함수 $\pi(x)$가 $\frac{x}{\ln x}$에 근접함을 추측했다.[1] 이를 통해서 알 수 있는 사실은, 어떤 수 n까지의 소수를 알아내는 풀이 1에 의한 계산에서 약 $\ln n$배의 개수에 해당하는 수를 담는 메모리를 사전에 구비해야 한다는 것이다.

실제로 100까지의 소수의 개수를 알아내는 세션에서 발견된 소수의 개수는 25개였으므로 실제 소수 개수의 4배에 해당하는 메모리를 필요로 했다.

풀이 2

풀이 1의 한계를 해결하기 위해서 다른 방식을 사용할 수 있다.

기본적으로 현재 확인하려는 수의 미만에 해당하는 수 중 소수의 목록을 알고 있다는 에라토스테네스의 체 알고리즘의 성질을 활용하여, 소수 목록만으로 수의 소수 여부를 확인할 수 있다.

2부터 시작하여, 소수 목록에 있는 수로 그 수를 나누었을 때에 하나라도 나누어떨어지는 것이 있다면 그 수는 소수가 아니라고 하고, 만약 그 수가 소수라면 소수 목록에 해당 수를 추가하는 과정을 상한까지 반복하면 소수 목록에는 상한까지의 소수들이 담겨있게 될 것이다.

처음에는 2에 대해서 계산을 진행한다. 처음에는 소수 목록에 아무런 수도 담겨있지 않기 때문에 2는 소수 목록에 들어가고 소수 목록은 $\{2\}$가 된다.

다음은 3에 대해서 계산을 진행한다. 소수 목록에 있는 어떤 수로도 3은 나누어떨어지지 않기 때문에 소수 목록에 3을 추가하고 소수 목록은 $\{2, 3\}$이 된다.

다음은 4에 대해서 계산을 진행한다. 4는 2로 나누어떨어지기 때문에 소수 목록에 4를 추가하지 않는다.

다음은 5에 대해서 계산을 진행한다. 소수 목록에 있는 어떤 수로도 5는 나누어떨어지지 않기 때문에 소수 목록에 5를 추가하고 소수 목록은 $\{2, 3, 5\}$가 된다.

이와 같은 과정을 상한인 100까지 반복하여 얻어진 소수 목록은 {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}이 된다.

이것을 Python 3 프로그래밍 언어로 구현하면 다음과 같다.

```python
def something_divisible(i: int, primes: list) -> bool:
  """ ``i``가 ``primes``에 있는 어떤 수로 나누어 떨어지는지 확인한다. """
  for prime in primes:
    if i % prime == 0:
      return True
  return False


def eratosthenes(limit: int) -> list:
  """ 에라토스테네스의 체 방식으로 ``limit``까지의 소수를 구해 출력한다. """
  primes = list()

  for i in range(2, limit+1):
    if not something_divisible(i, primes):
      primes.append(i)

  return primes
```

```
if __name__ == '__main__':
  print(eratosthenes(100))
```

이 방식은 컴퓨터 알고리즘을 만들 때에 자주 사용된다. 풀이 1에서 서술한 방법에 비해 메모리 사용량이 적기 때문이다.

#13+

$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$에 대해 가우스-조단 방법으로 역행렬을 해석적으로 구하라.

풀이

가우스-조단 방법은 정사각행렬 $\mathbf{A}$와 $\mathbf{A}$의 크기에 상응하는 단위행렬 $\mathbf{I}$에 대해 증가행렬 $\left(\begin{array}{c|c} \mathbf{A} & \mathbf{I} \end{array} \right)$를 상정하고, 기본 행 연산을 통해 $\left(\begin{array}{c|c} \mathbf{I} & \mathbf{A}^{-1} \end{array} \right)$의 형태로 만들어 행렬의 역행렬을 계산하는 방법이다.

따라서 일단 증가행렬

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 4 & 5 & 6 & 0 & 1 & 0 \\ 7 & 8 & 9 & 0 & 0 & 1 \end{array} \right)$$

을 상정하고, 2행 1열의 요소를 0으로 만들기 위해서 1행에 4를 곱한 것을 2행에서 빼주었다.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & -3 & -6 & -4 & 1 & 0 \\ 7 & 8 & 9 & 0 & 0 & 1 \end{array} \right)$$

이후 3행 1열의 요소를 0으로 만들기 위해서 1행에 7을 곱한 것을 3행에서 빼주었다.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & -3 & -6 & -4 & 1 & 0 \\ 0 & -6 & -12 & -7 & 0 & 1 \end{array} \right)$$

이후 3행 2열의 요소를 0으로 만들기 위해서 2행에 2를 곱한 것을 3행에서 빼주었다.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & -3 & -6 & -4 & 1 & 0 \\ 0 & 0 & 0 & -7 & -2 & 1 \end{array} \right)$$

이때, 3행 3열의 요소가 0이 되었으므로 가우스 조단 방법으로 역행렬을 구할 수 없게 된 것을 확인하고, 다른 조합으로 왼편을 단위행렬로 만들기 위해 시도했으나 실패했다.

이것이 풀이 과정 상의 문제일 것이라고 생각하지 않고, 문제 자체의 오류일 것이라고 생각하여, 사러스 법칙에 따라 행렬 $\mathbf{A}$의 행렬식을 계산했다.

$$\begin{aligned}
\det \mathbf{A} &= \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \\
&= 1 \cdot 5 \cdot 9 + 2 \cdot 6 \cdot 7 + 3 \cdot 4 \cdot 8 - 1 \cdot 6 \cdot 8 - 2 \cdot 4 \cdot 9 - 3 \cdot 5 \cdot 7 \\
&= 225 - 225 \\
&= 0
\end{aligned}$$

$\mathbf{A}$의 행렬식이 0이므로 역행렬이 정의되지 않는다. 따라서 가우스-조던 방법을 통해서 역행렬을 구할 수 없다.

#26(b)

$\mathbf{A} = \begin{pmatrix} 2 & 1 & 3 \\ 4 & 1 & -2 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 & 1 \\ 1 & 2 \\ 2 & 3 \end{pmatrix}$에 대해 $\mathbf{A} \cdot \mathbf{B}$와 $\mathbf{B} \cdot \mathbf{A}$를 구하라.

풀이

$$\begin{aligned}
\mathbf{AB} &= \begin{pmatrix} 2 \cdot 0 + 1 \cdot 1 + 3 \cdot 2 & 2 \cdot 1 + 1 \cdot 2 + 3 \cdot 3 \\ 4 \cdot 0 + 1 \cdot 1 + (-2) \cdot 2 & 4 \cdot 1 + 1 \cdot 2 + (-2) \cdot 3 \end{pmatrix} \\
&= \begin{pmatrix} 0 + 1 + 6 & 2 + 2 + 9 \\ 0 + 1 - 4 & 4 + 2 - 6 \end{pmatrix} \\
&= \begin{pmatrix} 7 & 13 \\ -3 & 0 \end{pmatrix}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{BA} &= \begin{pmatrix} 0\cdot 2+1\cdot 4 & 0\cdot 1+1\cdot 1 & 0\cdot 3+1(-2) \\ 1\cdot 2+2\cdot 4 & 1\cdot 1+2\cdot 1 & 1\cdot 3+2(-2) \\ 2\cdot 2+3\cdot 4 & 2\cdot 1+3\cdot 1 & 2\cdot 3+3(-2) \end{pmatrix} \\
&= \begin{pmatrix} 0+4 & 0+1 & 0-2 \\ 2+8 & 1+2 & 3-4 \\ 4+12 & 2+3 & 6-6 \end{pmatrix} \\
&= \begin{pmatrix} 4 & 1 & -2 \\ 10 & 3 & -1 \\ 16 & 5 & 0 \end{pmatrix}
\end{aligned}
$$

#29(b)

$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$에 대해 $\mathbf{A} \vee \mathbf{B}, \mathbf{A} \wedge \mathbf{B}, \mathbf{A} \odot \mathbf{B}$를 구하라.

풀이

행렬 $\mathbf{A}, \mathbf{B}$는 부울 행렬로, 부울 행렬만의 연산 $\vee$, $\wedge$, $\odot$이 정의된다. 각각은 접합, 교합, 부울곱 연산이다.

접합은 각각의 원소에 대해 논리합 연산을 수행한 것이다.

$$
\begin{aligned}
\mathbf{A} \vee \mathbf{B} &= \begin{pmatrix} 1\vee 1 & 0\vee 1 & 0\vee 1 \\ 0\vee 0 & 1\vee 0 & 1\vee 1 \\ 1\vee 1 & 0\vee 0 & 0\vee 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

교합은 각각의 원소에 대해 논리곱 연산을 수행한 것이다.

$$
\begin{aligned}
\mathbf{A} \wedge \mathbf{B} &= \begin{pmatrix} 1\wedge 1 & 0\wedge 1 & 0\wedge 1 \\ 0\wedge 0 & 1\wedge 0 & 1\wedge 1 \\ 1\wedge 1 & 0\wedge 0 & 0\wedge 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

행렬의 부울곱은 행렬곱에서 덧셈을 논리합으로, 곱셈을 논리곱으로 바꾼 것과 같다.

$$
\begin{aligned}
&\mathbf{A} \odot \mathbf{B} \\
&= \begin{pmatrix} 1 \wedge 1 \vee 0 \wedge 0 \vee 0 \wedge 1 & 1 \wedge 1 \vee 0 \wedge 0 \vee 0 \wedge 0 & 1 \wedge 1 \vee 0 \wedge 1 \vee 0 \wedge 1 \\ 0 \wedge 1 \vee 1 \wedge 0 \vee 1 \wedge 1 & 0 \wedge 1 \vee 1 \wedge 0 \vee 1 \wedge 0 & 0 \wedge 1 \vee 1 \wedge 1 \vee 1 \wedge 1 \\ 1 \wedge 1 \vee 0 \wedge 0 \vee 0 \wedge 1 & 1 \wedge 1 \vee 0 \wedge 0 \vee 0 \wedge 0 & 1 \wedge 1 \vee 0 \wedge 1 \vee 0 \wedge 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 \vee 0 \vee 0 & 1 \vee 0 \vee 0 & 1 \vee 0 \vee 0 \\ 0 \vee 0 \vee 1 & 0 \vee 0 \vee 0 & 0 \vee 1 \vee 1 \\ 1 \vee 0 \vee 0 & 1 \vee 0 \vee 0 & 1 \vee 0 \vee 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}
\end{aligned}
$$

#31(c)+

행렬 $\begin{pmatrix} 2 & 6 & 5 \\ 0 & 3 & 2 \\ 4 & 12 & 1 \end{pmatrix}$에 대해, 사러스 법칙에 의해 행렬식을 먼저 구하고, 다음으로는 소 행렬식과 여인수를 이용하여 행렬식을 구하라.

풀이

행렬 $\mathbf{A} = \begin{pmatrix} 2 & 6 & 5 \\ 0 & 3 & 2 \\ 4 & 12 & 1 \end{pmatrix}$의 사러스 법칙에 의한 행렬식은 다음과 같이 계산된다.

$$
\begin{aligned}
&\begin{vmatrix} 2 & 6 & 5 \\ 0 & 3 & 2 \\ 4 & 12 & 1 \end{vmatrix} \\
&= 2 \cdot 3 \cdot 1 + 6 \cdot 2 \cdot 4 + 5 \cdot 0 \cdot 12 \\
&- 2 \cdot 2 \cdot 12 - 6 \cdot 0 \cdot 1 - 5 \cdot 3 \cdot 4 \\
&= 6 + 48 + 0 - 48 - 0 - 60 \\
&= -54
\end{aligned}
$$

위 행렬의 소행렬은 다음과 같다.

$$\begin{pmatrix} \begin{vmatrix} 3 & 2 \\ 12 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 2 \\ 4 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 3 \\ 4 & 12 \end{vmatrix} \\ \begin{vmatrix} 6 & 5 \\ 12 & 1 \end{vmatrix} & \begin{vmatrix} 2 & 5 \\ 4 & 1 \end{vmatrix} & \begin{vmatrix} 2 & 6 \\ 4 & 12 \end{vmatrix} \\ \begin{vmatrix} 6 & 5 \\ 3 & 2 \end{vmatrix} & \begin{vmatrix} 2 & 5 \\ 0 & 2 \end{vmatrix} & \begin{vmatrix} 2 & 6 \\ 0 & 3 \end{vmatrix} \end{pmatrix}$$

이때, 임의의 2×2행렬 $\mathbf{M} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$의 소행렬은 $\begin{pmatrix} d & c \\ b & a \end{pmatrix}$이고 $\mathbf{M}$의 여인수 행렬은 $\begin{pmatrix} d & -c \\ -b & a \end{pmatrix}$이므로 $\mathbf{M}$의 행렬식은 $ad - bc$인 것을 알 수 있다. 이것에 근거하여 $\mathbf{A}$의 소행렬을 계산하면 다음과 같다.

$$\begin{aligned} &\begin{pmatrix} 3 \cdot 1 - 2 \cdot 12 & 0 \cdot 1 - 2 \cdot 4 & 0 \cdot 12 - 3 \cdot 4 \\ 6 \cdot 1 - 5 \cdot 12 & 2 \cdot 1 - 5 \cdot 4 & 2 \cdot 12 - 6 \cdot 4 \\ 6 \cdot 2 - 5 \cdot 3 & 2 \cdot 2 - 5 \cdot 0 & 2 \cdot 3 - 6 \cdot 0 \end{pmatrix} \\ = &\begin{pmatrix} -21 & -8 & -12 \\ -54 & -18 & 0 \\ -3 & 4 & 6 \end{pmatrix} \end{aligned}$$

이 소행렬에 근거하여 $\mathbf{A}$의 여인수 행렬을 계산하면 다음과 같다.

$$\begin{pmatrix} -21 & 8 & -12 \\ 54 & -18 & 0 \\ -3 & -4 & 6 \end{pmatrix}$$

이 여인수 행렬에 근거하여 1행에서 $\mathbf{A}$의 행렬식을 계산하면 다음과 같다.

$$\begin{aligned} \det \mathbf{A} &= 2(-21) + 6 \cdot 8 + 5(-12) \\ &= -42 + 48 - 60 \\ &= -54 \end{aligned}$$

#33

행렬 $\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$에 대해 고유값 및 고유벡터를 구하라.

풀이

행렬 $\mathbf{A}$에 대한 고유값 λ와 고유벡터 $\mathbf{x}$ $(\mathbf{x} \neq \mathbf{0})$는 방정식 $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$를 만족하는 λ와 $\mathbf{x}$를 말한다. λ를 구하기 위해서는 $\mathbf{A}$에 대한 특성방정식 $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$의 해를 구해야 한다.

$\mathbf{A} - \lambda\mathbf{I}$에 대한 행렬식은 다음과 같이 주어진다.

$$\begin{vmatrix} 2-\lambda & 1 & 1 \\ 1 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^3 + 1 + 1 - 3(2-\lambda) = 0$$

이때, $(2-\lambda)$를 Λ로 치환하여 주어진 삼차방정식을 해결한다.

$$\begin{aligned} (2-\lambda)^3 - 3(2-\lambda) + 2 = 0 \\ \Lambda^3 - 3\Lambda + 2 = 0 \\ (\Lambda - 1)(\Lambda^2 + \Lambda - 2) = 0 \\ (\Lambda - 1)^2(\Lambda + 2) = 0 \end{aligned}$$

따라서 $\Lambda = 1, -2$이고, $\lambda = 1, 4$이다.

$\lambda = 1$인 경우, 고유벡터 $\mathbf{x}$의 요소 x_i에 대한 연립방정식으로 다음과 같은 증가 행렬이 주어진다.

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array}\right)$$

따라서 $x_1 + x_2 + x_3 = 0$인 모든 벡터가 $\lambda = 1$일 때 $\mathbf{A}$의 고유벡터임을 알 수 있다. 이와 같은 벡터에는 $\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$ 등을 들 수 있다.

$\lambda = 4$인 경우, 고유벡터 $\mathbf{x}$의 요소 x_i에 대한 연립방정식으로 다음과 같은 증가 행렬이 주어진다.

$$\left(\begin{array}{ccc|c} -2 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -2 & 0 \end{array}\right)$$

위 행렬의 1행과 2행을 교환하면 다음과 같다.

$$\left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ -2 & 1 & 1 & 0 \\ 1 & 1 & -2 & 0 \end{array}\right)$$

위 행렬의 2행에서 1행에 2를 곱해 더하고, 3행에서 1행을 빼면 다음과 같다.

$$\left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 3 & -3 & 0 \end{array}\right)$$

위 행렬의 3행에서 2행을 더하고, 2행을 -3으로 나누면 다음과 같다.

$$\left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

따라서

$$\begin{aligned} x_2 - x_3 &= 0 \\ x_2 &= x_3 \end{aligned}$$

임을 알 수 있으며,

$$\begin{aligned} x_1 - 2x_2 + 1 &= 0 \\ x_1 + (-2+1)x_2 &= 0 \\ x_1 - x_2 &= 0 \\ x_1 &= x_2 \end{aligned}$$

임 역시 알 수 있다. 따라서 $x_1 = x_2 = x_3$인 모든 벡터가 $\lambda = 4$일 때 $\mathbf{A}$의 고유벡터임을 알 수 있다. 이와 같은 벡터에는 $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ 등을 들 수 있다.

참고 문헌

1. ko.wikipedia.org, 소수 계량 함수 (https://ko.wikipedia.org/wiki/소수_계량_함수), 2023년 3월 28일 방문