시리즈 마무리

이 장의 목적:

- 선형대수의 본질 재고찰
- "벡터"란 무엇인가?
- 추상화의 힘

핵심 질문:

```
벡터 = 화살표?
벡터 = 숫자 리스트?
벡터 = ???
```

지금까지의 여정

우리가 배운 것:

- 벡터: 공간의 화살표
- 행렬: 선형변환
- 내적, 외적, 행렬식
- 고유벡터, 고유값

시각적 직관:

```
모든 개념을 기하학적으로 이해
```

좌표계의 임의성

중요한 관찰:

행렬식:

```
det(A) = 넓이/부피 스케일링 계수
```

좌표계를 바꿔도:

```
det(A) = 변하지 않음
```

고유벡터:

```
변환 후 자신의 span에 남는 벡터
```

좌표계를 바꿔도:

```
고유벡터 = 같은 방향
고유값 = 같은 값
```

관찰:

```
핵심 성질들이 좌표와 무관!
```

벡터의 본질

질문:

```
벡터가 근본적으로 숫자 리스트가 아니라면,
그 본질은 무엇인가?
```

단서:

```
핵심 성질들이 공간적(spatial)
```

함의:

```
벡터 = 더 근본적인 공간적 존재
좌표 = 그것을 표현하는 방법
```

예제: 함수

놀라운 사실:

```
함수도 "벡터처럼" 행동!
```

함수의 덧셈:

```
(f + g)(x) = f(x) + g(x)
```

예:

```
f(x) = x²
g(x) = 2x

(f + g)(x) = x² + 2x
```

함수의 스케일링:

```
(c·f)(x) = c·f(x)
```

예:

```
f(x) = sin(x)
(3f)(x) = 3sin(x)
```

화살표처럼:

```
함수를 더하고 스케일할 수 있음!
```

함수의 선형변환

미분:

```
d/dx: 함수 → 함수
```

선형성:

```
d/dx(f + g) = d/dx(f) + d/dx(g)
d/dx(c·f) = c·d/dx(f)
```

예:

```
f(x) = x³
g(x) = 2x

d/dx(x³ + 2x) = d/dx(x³) + d/dx(2x)
                = 3x² + 2
```

선형변환!

용어:

```
함수의 경우: "연산자(operator)"
벡터의 경우: "변환(transformation)"

의미는 같음
```

선형변환의 정의 (재고찰)

기하학적 정의 (화살표용):

```
1. 직선 → 직선
2. 격자선 평행 유지
3. 원점 고정
```

하지만 함수는:

```
"격자선"이 뭐지?
```

더 일반적 정의 필요!

대수적 정의:

```
L(v + w) = L(v) + L(w)  (가법성)
L(c·v) = c·L(v)         (동차성)
```

이것은:

```
화살표에도 적용 ✓
함수에도 적용 ✓
다른 것들에도? ✓
```

벡터공간의 공리

벡터공간 V:

연산 1: 덧셈

```
v, w ∈ V → v + w ∈ V
```

연산 2: 스칼라 곱

```
c ∈ ℝ, v ∈ V → c·v ∈ V
```

공리 (8개):

덧셈 관련:

1. **교환법칙:** $v + w = w + v$
2. **결합법칙:** $(u + v) + w = u + (v + w)$
3. **항등원:** $0 + v = v$ (영벡터 존재)
4. **역원:** $v + (-v) = 0$ (음벡터 존재)

스칼라 곱 관련: 5. **분배법칙 1:** $c(v + w) = cv + cw$ 6. **분배법칙 2:** $(a + b)v = av + bv$ 7. **결합법칙:** $a(bv) = (ab)v$ 8. **항등원:** $1 \cdot v = v$

이것만 만족하면:

```
"벡터공간"!
```

벡터공간의 예

1. $\mathbb{R}^n$ (숫자 리스트):

```
v = [v₁, v₂, ..., vₙ]
```

덧셈:

```
[v₁, v₂] + [w₁, w₂] = [v₁+w₁, v₂+w₂]
```

스칼라 곱:

```
c[v₁, v₂] = [cv₁, cv₂]
```

모든 공리 만족 ✓

---

2. 함수 공간:

```
V = {모든 실함수}
```

덧셈:

```
(f + g)(x) = f(x) + g(x)
```

스칼라 곱:

```
(cf)(x) = c·f(x)
```

영벡터:

```
0(x) = 0 (영함수)
```

모든 공리 만족 ✓

---

3. 다항식 공간:

```
P_n = {차수 ≤ n인 다항식}
```

예 (P_2):

```
p(x) = a₀ + a₁x + a₂x²
```

덧셈:

```
(p + q)(x) = p(x) + q(x)
```

닫혀있음:

```
차수 ≤ 2 + 차수 ≤ 2 = 차수 ≤ 2 ✓
```

모든 공리 만족 ✓

4. 행렬 공간:

```
M_{m×n} = {m×n 행렬}
```

덧셈:

```
[a  b] + [e  f] = [a+e  b+f]
[c  d]   [g  h]   [c+g  d+h]
```

스칼라 곱:

```
k[a  b] = [ka  kb]
 [c  d]   [kc  kd]
```

모든 공리 만족 ✓

선형변환의 재정의

일반적 정의:

```
T: V → W (벡터공간 V에서 W로)
```

선형조건:

```
T(v + w) = T(v) + T(w)
T(cv) = c·T(v)
```

예제:

미분 연산자:

```
D: P₃ → P₂
D(a₀ + a₁x + a₂x² + a₃x³) = a₁ + 2a₂x + 3a₃x²
```

선형성:

```
D(p + q) = D(p) + D(q) ✓
D(cp) = c·D(p) ✓
```

적분 연산자:

```
I: P_n → P_{n+1}
I(p(x)) = ∫₀ˣ p(t) dt
```

선형성:

```
I(p + q) = I(p) + I(q) ✓
I(cp) = c·I(p) ✓
```

선형대수 개념의 일반화

지금까지 배운 모든 것:

1. 선형변환

- 화살표: 공간 변환
- 함수: 연산자

2. 고유벡터/고유값

```
T(v) = λv
```

미분 예:

```
d/dx(eˣ) = eˣ

eˣ는 미분의 고유함수!
고유값 = 1
```

3. 기저

```
span하고 선형독립인 벡터들
```

다항식:

```
P₂의 기저: {1, x, x²}
```

4. 차원

```
dim(P₂) = 3
```

5. 내적

```
함수의 내적:
⟨f, g⟩ = ∫ f(x)g(x) dx
```

6. 직교성

```
⟨f, g⟩ = 0
```

푸리에 급수:

- sin, cos 함수들

- 서로 직교
- 함수공간의 "기저"

좌표의 재해석

화살표 벡터:

```
v = [3, 2]
```

의미:

```
v = 3î + 2ĵ (기저의 선형결합)
```

다항식:

```
p(x) = 2 + 3x + 5x²
```

기저 {1, x, x²}에서:

```
좌표 = [2, 3, 5]
```

의미:

```
p = 2·1 + 3·x + 5·x²
```

일반적:

```
좌표 = 기저에 대한 계수
```

선형변환과 행렬

유한차원 벡터공간:

기저 선택:

```
V: 기저 {v₁, v₂, ..., vₙ}
W: 기저 {w₁, w₂, ..., wₘ}
```

선형변환 T: V → W

행렬 표현:

```
각 기저벡터 vᵢ를 T로 변환
→ W의 기저로 좌표 표현
→ 행렬의 i번째 열
```

예: 미분

```
D: P₂ → P₁
기저: P₂ = {1, x, x²}, P₁ = {1, x}
```

변환:

```
D(1) = 0 = 0·1 + 0·x → [0, 0]
D(x) = 1 = 1·1 + 0·x → [1, 0]
D(x²) = 2x = 0·1 + 2·x → [0, 2]
```

행렬:

```
[0  1  0]
[0  0  2]
```

적용:

```
p(x) = 2 + 3x + 5x²

좌표: [2]
      [3]
      [5]

D(p) 좌표: [0  1  0] [2]   [3 ]
           [0  0  2] [3] = [10]
                     [5]

D(p) = 3·1 + 10·x = 3 + 10x ✓
```

추상화의 힘

왜 이렇게 일반화?

1. 통일된 이론:

```
한 번 증명 → 모든 곳에 적용
```

예:

- 행렬식 이론
- 고유값 계산
- 대각화

모두 추상 공간에도!

2. 새로운 통찰:

```
서로 다른 분야의 연결
```

예:

- 미분방정식 ↔ 선형대수
- 푸리에 급수 ↔ 기저 변환
- 양자역학 ↔ 힐베르트 공간

3. 계산 도구:

```
추상 문제 → 행렬 문제
```

미분방정식:

```
dy/dx = Ay

행렬 지수: e^(Ax)
고유벡터로 풀이
```

벡터란 무엇인가? (최종 답)

세 가지 관점:

1. 물리학자:

```
벡터 = 공간의 화살표
```

2. 컴퓨터 과학자:

```
벡터 = 숫자 리스트
```

3. 수학자:

```
벡터 = 덧셈과 스칼라 곱이 정의된 대상
```

통합:

```
벡터 = 벡터공간의 원소
```

본질:

```
좌표와 무관한 기하학적/대수적 구조
```

선형대수의 핵심

도구들:

- 선형변환
- 영공간
- 고유벡터
- 내적

적용 범위:

```
화살표? ✓
숫자 리스트? ✓
함수? ✓
행렬? ✓
당신이 정의한 이상한 것? ✓
```

조건:

```
덧셈과 스칼라 곱이 합리적으로 정의되면 OK
```

추상 사고의 가치

수학자의 관점:

목표:

```
모든 벡터적 대상에 적용되는 이론
```

방법:

```
공통된 성질 추출 (8개 공리)
```

결과:

```
한 번 증명 = 무한한 적용
```

예:

```
"선형독립 벡터의 최대 집합 = 기저"

증명: 추상 벡터공간에서 한 번
적용: 화살표, 함수, 행렬, ...
```

실용적 의미

응용 분야:

1. 미분방정식:

```
해공간 = 벡터공간
선형 연산자 이론 적용
```

2. 양자역학:

```
상태 = 힐베르트 공간의 벡터
관측 = 선형 연산자
```

3. 신호처리:

```
신호 = 함수공간의 벡터
푸리에 변환 = 기저 변환
```

4. 기계학습:

```
데이터 = 고차원 벡터
PCA = 고유벡터 분석
```

5. 컴퓨터 그래픽스:

```
다항식 곡선 = 벡터공간
변환 = 행렬 연산
```

핵심 직관 정리

1. 벡터의 본질:

```
좌표 = 표현 방법
벡터 = 그 자체
```

2. 공리적 정의:

```
8개 공리 만족 → 벡터공간
```

3. 선형변환:

```
T(v + w) = T(v) + T(w)
T(cv) = c·T(v)

이것만 있으면 선형!
```

4. 모든 도구 일반화:

```
기저, 차원, 내적, 고유벡터, ...
→ 모든 벡터공간에!
```

5. 추상화의 힘:

```
한 번 배우면 → 어디든 적용
```

시리즈를 마치며

배운 것:

- 벡터의 기하학적 직관

- 선형변환의 본질
- 행렬식, 내적, 외적
- 고유벡터/고유값
- 추상화

핵심 메시지:

```
선형대수 = 패턴의 언어
```

어디서나:

- 공간 변환
- 함수 연산
- 데이터 분석
- 물리 법칙

모두 선형대수로 표현!

다음 단계:

```
이 직관을 가지고:
- 더 깊은 이론 학습
- 실제 문제 적용
- 새로운 연결 발견
```

수학을 사랑하세요!

---

2. 퀴즈

문제 1

벡터공간의 정의를 서술하시오. 특히: (a) 벡터공간이 갖춰야 할 두 가지 연산은 무엇인가? (b) 8개의 공리를 모두 나열하시오 (덧셈 4개, 스칼라 곱 4개). (c) 왜 이러한 공리들이 필요한지, 구체적 예를 들어 설명하시오.

문제 2

함수가 벡터공간을 이룬다는 것을 설명하시오: (a) 함수의 덧셈과 스칼라 곱을 정의하시오. (b) 미분 연산자 d/dx가 선형변환임을 증명하시오. (c) 지수함수 e^x가 미분 연산자의 "고유함수"임을 보이고, 고유값을 구하시오.

문제 3

다항식 공간 $P_2 = \{a_0 + a_1x + a_2x^2 : a_0, a_1, a_2 \in \mathbb{R}\}$에 대해: (a) 표준 기저를 제시하고 차원을 구하시오. (b) 다항식 $p(x) = 2 + 3x + 5x^2$의 좌표를 구하시오. (c) 미분 연산자 $D: P_2 \to P_1$의 행렬 표현을 구하시오 (P_1의 기저는 $\{1, x\}$).

문제 4

"벡터란 무엇인가?"라는 질문에 대한 세 가지 관점을 비교하시오: (a) 물리학자의 관점 (화살표) (b) 컴퓨터 과학자의 관점 (숫자 리스트) (c) 수학자의 관점 (추상 벡터공간) 그리고 이 세 관점이 어떻게 통합되는지 설명하시오.

문제 5

추상화의 가치를 설명하시오: (a) 왜 수학자들은 벡터를 추상적으로 정의하려 하는가? (b) 구체적인 예를 들어, 추상 벡터 공간 이론이 어떻게 서로 다른 분야를 연결하는지 서술하시오. (c) 선형대수의 개념들(기저, 선형변환, 고유벡터 등)이 함수공간에도 적용되는 이유를 공리적 관점에서 설명하시오.

---

3. 퀴즈 답

문제 1 답안

벡터공간의 정의:

벡터공간 V:

```
집합 + 두 연산 + 8개 공리
```

(a) 두 가지 연산:

연산 1: 벡터 덧셈

```
+ : V × V → V
(v, w) ↦ v + w
```

의미:

- 두 벡터를 더하면
- 또 다른 벡터

닫혀있음 (closure):

```
v, w ∈ V → v + w ∈ V
```

연산 2: 스칼라 곱

```
· : ℝ × V → V
(c, v) ↦ c·v
```

의미:

- 벡터를 스칼라로 곱하면
- 또 다른 벡터

닫혀있음:

```
c ∈ ℝ, v ∈ V → c·v ∈ V
```

(b) 8개 공리:

벡터 덧셈 공리 (4개):

A1. 교환법칙 (Commutativity):

```
v + w = w + v  (∀v, w ∈ V)
```

A2. 결합법칙 (Associativity):

```
(u + v) + w = u + (v + w)  (∀u, v, w ∈ V)
```

A3. 항등원 (Additive Identity):

```
∃ 0 ∈ V such that v + 0 = v  (∀v ∈ V)
```

"영벡터" 존재

A4. 역원 (Additive Inverse):

```
∀v ∈ V, ∃ (-v) ∈ V such that v + (-v) = 0
```

모든 벡터는 "음벡터" 가짐

스칼라 곱 공리 (4개):

M1. 분배법칙 (벡터에 대해):

```
c(v + w) = cv + cw  (∀c ∈ ℝ, ∀v, w ∈ V)
```

M2. 분배법칙 (스칼라에 대해):

```
(a + b)v = av + bv  (∀a, b ∈ ℝ, ∀v ∈ V)
```

M3. 결합법칙:

```
a(bv) = (ab)v  (∀a, b ∈ ℝ, ∀v ∈ V)
```

M4. 항등원:

```
1·v = v  (∀v ∈ V)
```

1로 곱하면 변화 없음

(c) 공리의 필요성:

왜 이런 공리들?

목표:

```
"벡터처럼 행동"하는 것 포착
```

구체적 예:

교환법칙 (A1):

```
v + w = w + v
```

필요성:

- 덧셈 순서 무관
- 화살표: 평행사변형 대칭
- 숫자: 당연함

없다면:

```
v + w ≠ w + v

이상한 "덧셈"
벡터적 직관 깨짐
```

항등원 (A3):

```
0 + v = v
```

필요성:

- "아무것도 안 더하기" 개념
- 화살표: 영벡터
- 함수: 영함수 f(x) = 0

없다면:

```
"기준점" 없음
```

```
방정식 v + x = v의 해가 없을 수도
```

분배법칙 (M1):

```
c(v + w) = cv + cw
```

필요성:

- 선형성의 핵심
- "스케일"과 "덧셈"의 호환

예 (화살표):

```
2(v + w) = 2v + 2w

기하학적으로 자연스러움
```

없다면:

```
2(v + w) ≠ 2v + 2w

선형변환 정의 불가능
```

항등원 (M4):

```
1·v = v
```

필요성:

- 1로 곱하면 변화 없음
- "스케일 안 하기"

없다면:

```
1·v ≠ v

v를 "자기 자신"으로 표현 못함
```

전체적 필요성:

1. 일관성:

```
모든 벡터 개념이 이 공리에서 도출
```

2. 최소성:

```
8개면 충분
더 적으면 부족
```

3. 범용성:

```
이 공리만 만족하면
→ 모든 선형대수 이론 적용 가능
```

결론: 공리는 "벡터다움"의 정확한 수학적 정의입니다!

문제 2 답안

함수의 벡터공간:

집합:

```
V = {f : ℝ → ℝ}  (모든 실함수)
```

(a) 연산 정의:

덧셈:

```
(f + g)(x) = f(x) + g(x)
```

의미:

- 각 점 x에서
- 함수값을 더함

예:

```
f(x) = x²
g(x) = 2x

(f + g)(x) = x² + 2x
```

그래프적:

```
f + g의 y값 = f의 y값 + g의 y값
```

스칼라 곱:

```
(c·f)(x) = c·f(x)
```

의미:

- 각 점 x에서
- 함수값을 c배

예:

```
f(x) = sin(x)

(3f)(x) = 3sin(x)
```

그래프적:

```
3f = f를 수직으로 3배 늘림
```

영벡터:

```
0(x) = 0  (모든 x)
```

영함수

역원:

```
(−f)(x) = −f(x)
```

8개 공리 확인:

A1: f + g = g + f

```
(f + g)(x) = f(x) + g(x) = g(x) + f(x) = (g + f)(x) ✓
```

(나머지도 모두 점별로 확인 가능)

결론: 함수는 벡터공간! ✓

(b) 미분 연산자의 선형성:

정의:

```
D : V → V
D(f) = df/dx
```

증명해야 할 것:

선형성 조건 1: 가법성

```
D(f + g) = D(f) + D(g)
```

증명:

```
D(f + g) = d/dx(f + g)
         = d/dx(f(x) + g(x))

미분의 가법성:
         = d/dx(f(x)) + d/dx(g(x))
         = D(f) + D(g) ✓
```

예:

```
f(x) = x³, g(x) = sin(x)

D(x³ + sin(x)) = D(x³) + D(sin(x))
               = 3x² + cos(x) ✓
```

선형성 조건 2: 동차성

```
D(c·f) = c·D(f)
```

증명:

```
D(c·f) = d/dx(c·f)
       = d/dx(c·f(x))

미분의 동차성:
       = c·d/dx(f(x))
       = c·D(f) ✓
```

예:

```
f(x) = x²

D(5x²) = 5·D(x²)
       = 5·2x
       = 10x ✓
```

결론:

```
D(f + g) = D(f) + D(g) ✓
D(c·f) = c·D(f) ✓

→ D는 선형변환!
```

(c) 지수함수의 고유함수:

정의:

```
f(x) = eˣ
```

미분:

```
D(eˣ) = d/dx(eˣ) = eˣ
```

관찰:

```
D(eˣ) = eˣ = 1·eˣ
```

고유함수 조건:

```
D(f) = λf
```

적용:

```
D(eˣ) = 1·eˣ

λ = 1
```

결론:

```
eˣ는 미분 연산자 D의 고유함수
고유값 = 1
```

더 일반적:

f(x) = e^(λx):

```
D(e^(λx)) = d/dx(e^(λx))
          = λe^(λx)
          = λ·e^(λx)
```

고유함수!

```
고유값 = λ
```

예:

```
f(x) = e^(3x)

D(e^(3x)) = 3e^(3x) = 3·f(x)

고유값 = 3
```

물리적 의미:

미분방정식:

```
dy/dx = y

해: y = Ceˣ
```

고유함수 관점:

```
Dy = 1·y

y는 고유값 1의 고유함수
→ y = Ceˣ
```

일반화:

```
dy/dx = λy

해: y = Ce^(λx)

고유함수 이론으로 자연스럽게 설명!
```

이것이 선형대수와 미분방정식의 연결입니다!

문제 3 답안

다항식 공간 P_2:

정의:

```
P₂ = {a₀ + a₁x + a₂x² : a₀, a₁, a₂ ∈ ℝ}
```

차수가 2 이하인 모든 다항식

(a) 표준 기저와 차원:

기저 후보:

```
B = {1, x, x²}
```

확인해야 할 것:

1. Span:

임의의 $p \in P_2$:

```
p(x) = a₀ + a₁x + a₂x²
     = a₀·1 + a₁·x + a₂·x²
```

B의 선형결합으로 표현 가능 ✓

2. 선형독립:

```
c₀·1 + c₁·x + c₂·x² = 0 (영다항식)
```

이면:

```
c₀ = 0, c₁ = 0, c₂ = 0
```

왜? 다항식의 계수 비교:

```
c₀ + c₁x + c₂x² = 0 (항등적으로)

x⁰ 계수: c₀ = 0
x¹ 계수: c₁ = 0
x² 계수: c₂ = 0
```

유일해 → 선형독립 ✓

결론:

```
기저: {1, x, x²}
차원: dim(P₂) = 3
```

다른 기저 예:

```
{1, 1+x, 1+x+x²}
{1−x, x+x², x²}
```

선형독립이고 span하면 기저!

(b) 다항식의 좌표:

주어진:

```
p(x) = 2 + 3x + 5x²
```

기저 B = {1, x, x²}에서:

선형결합:

```
p(x) = 2·1 + 3·x + 5·x²
```

좌표:

```
[p]_B = [2]
        [3]
        [5]
```

의미:

```
첫 번째 성분: 1의 계수 = 2
두 번째 성분: x의 계수 = 3
세 번째 성분: x²의 계수 = 5
```

다른 기저라면:

기저 $B' = \{1, 1+x, x^2\}$

```
p(x) = 2 + 3x + 5x²
     = ?·1 + ?·(1+x) + ?·x²
```

계산:

```
= a·1 + b·(1+x) + c·x²
= (a+b) + bx + cx²

a+b = 2
b = 3
c = 5

→ a = -1, b = 3, c = 5
```

좌표:

```
[p]_B' = [-1]
         [ 3]
         [ 5]
```

같은 다항식, 다른 좌표!

(c) 미분 연산자의 행렬 표현:

연산자:

```
D : P₂ → P₁
D(a₀ + a₁x + a₂x²) = a₁ + 2a₂x
```

기저:

```
P₂: B₂ = {1, x, x²}
P₁: B₁ = {1, x}
```

Step 1: 각 기저벡터 변환

D(1):

```
D(1) = 0 = 0·1 + 0·x
```

B_1에서 좌표:

```
[D(1)]_{B₁} = [0]
              [0]
```

D(x):

```
D(x) = 1 = 1·1 + 0·x
```

B_1에서 좌표:

```
[D(x)]_{B₁} = [1]
              [0]
```

D(x²):

```
D(x²) = 2x = 0·1 + 2·x
```

B_1에서 좌표:

```
[D(x²)]_{B₁} = [0]
               [2]
```

Step 2: 행렬 구성

각 열 = 변환된 기저벡터의 좌표:

```
[D] = [D(1)  D(x)  D(x²)]

    = [0  1  0]
      [0  0  2]
```

2×3 행렬 (P_1은 2차원, P_2는 3차원)

Step 3: 검증

p(x) = 2 + 3x + 5x²:

좌표:

```
[p]_{B₂} = [2]
           [3]
           [5]
```

행렬 적용:

```
[D][p]_{B₂} = [0  1  0] [2]   [3 ]
              [0  0  2] [3] = [10]
                        [5]
```

결과 해석:

```
[3, 10]_{B₁} = 3·1 + 10·x = 3 + 10x
```

확인:

```
D(2 + 3x + 5x²) = 3 + 10x ✓
```

결론:

미분 연산자의 행렬:

```
[0  1  0]
[0  0  2]
```

이것은 2×3 "비정방" 행렬!

- 3D (P_2) → 2D (P_1)

문제 4 답안

"벡터란 무엇인가?" - 세 가지 관점:

(a) 물리학자의 관점: 화살표

정의:

```
벡터 = 공간의 화살표
```

특징:

1. 방향과 크기:

```
크기: |v|
방향: v/|v|
```

2. 위치 무관:

```
평행이동해도 같은 벡터
```

3. 덧셈:

```
평행사변형 법칙
```

4. 스케일링:

```
화살표를 늘리거나 줄임
```

장점:

- 직관적
- 시각화 가능
- 물리적 의미 명확 (힘, 속도, ...)

단점:

- 3차원 이상 시각화 어려움
- 추상적 개념 표현 제한

(b) 컴퓨터 과학자의 관점: 숫자 리스트

정의:

```
벡터 = 순서 있는 숫자 리스트
```

예:

```
v = [3, -2, 5]
```

특징:

1. 구체적:

```
메모리에 저장 가능
```

2. 계산 가능:

```
덧셈: [1, 2] + [3, 4] = [4, 6]
스케일: 2[1, 2] = [2, 4]
```

3. 고차원 OK:

```
[x₁, x₂, ..., x₁₀₀₀]
```

4. 데이터로 해석:

```
[키, 몸무게, 나이]
[픽셀₁, 픽셀₂, ...]
```

장점:

- 계산 가능
- 프로그래밍 가능
- 고차원 처리

단점:

- 좌표계 의존
- 기하학적 직관 약함

(c) 수학자의 관점: 추상 벡터공간

정의:

```
벡터 = 벡터공간 V의 원소
```

벡터공간:

```
덧셈과 스칼라 곱이 정의되고
8개 공리를 만족하는 집합
```

특징:

1. 일반성:

```
화살표 ✓
숫자 리스트 ✓
함수 ✓
행렬 ✓
...
```

2. 본질 추구:

```
좌표와 무관한 성질
```

3. 공리적 접근:

```
정의로부터 모든 성질 도출
```

장점:

- 최대 일반성
- 통일된 이론
- 새로운 연결 발견

단점:

- 추상적
- 초기 학습 어려움

세 관점의 통합:

계층 구조:

```
수학자 (가장 일반)
    ↑
    |
물리학자, CS (특수 경우)
```

관계:

1. 화살표 → 숫자 리스트:

```
좌표계 선택
v = 3î + 2ĵ → [3, 2]
```

2. 숫자 리스트 → 추상 벡터:

```
ℝⁿ는 벡터공간의 예
[a, b, c] ∈ ℝ³
```

3. 화살표 → 추상 벡터:

```
기하학적 벡터 = 좌표와 무관한 존재
= 추상 벡터의 구체적 실현
```

통합된 이해:

벡터의 본질:

```
덧셈과 스칼라 곱이 "합리적"으로 정의된 대상
```

표현 방법:

- 물리: 화살표로 시각화
- CS: 좌표로 계산
- 수학: 공리로 추상화

모두 같은 것:

```
화살표 = 숫자 리스트 = 추상 벡터

관점만 다름
```

예시로 정리:

벡터 v:

물리학자:

```
→ (오른쪽 위 방향 화살표)
```

CS:

```
[3, 2]
```

수학자:

```
v ∈ ℝ² such that
v = 3e₁ + 2e₂
(e₁, e₂는 기저)
```

모두 같은 벡터를 다르게 표현!

핵심 통찰:

```
벡터 ≠ 표현
벡터 = 그 자체

좌표 = 특정 기저에서의 표현
본질 = 좌표와 무관
```

이것이 추상 벡터공간의 핵심입니다!

문제 5 답안

추상화의 가치:

(a) 왜 추상적 정의?

수학자의 목표:

```
한 번 증명 → 무한한 적용
```

동기:

1. 패턴 인식:

서로 다른 대상들:

```
- 화살표
- 숫자 리스트
- 함수
- 행렬
```

공통점 발견:

```
모두 "덧셈"과 "스케일링" 가능
```

질문:

```
이들의 공통된 본질은?
```

---

2. 중복 제거:

추상화 전:

```
화살표의 선형독립 증명
함수의 선형독립 증명
행렬의 선형독립 증명
...
```

비효율!

추상화 후:

```
추상 벡터공간의 선형독립 증명 (1번)
→ 모든 경우에 적용
```

효율!

---

3. 새로운 발견:

추상 이론:

```
정리: 선형독립 벡터의 최대 개수 = 차원
```

자동으로 적용:

```
화살표: 3D → 최대 3개
함수 (P₂): 최대 3개
2×2 행렬: 최대 4개
```

새로운 통찰 즉시 얻음!

---

4. 일반성:

구체적 정의:

```
특정 경우에만 작동
```

추상적 정의:

```
모든 경우에 작동
```

예:

```
"det(A) = 0 ↔ 특이"

추상화: "T: V → V가 가역 ↔ 단사/전사"
→ 무한차원에도 적용!
```

결론:

추상화는:

1. **효율성** (한 번 증명)
2. **일반성** (모든 곳에 적용)
3. **통찰** (새로운 연결 발견)
4. **아름다움** (본질 포착)

(b) 서로 다른 분야의 연결:

예 1: 미분방정식 ↔ 선형대수

미분방정식:

```
dy/dx = Ay  (A는 행렬)
```

추상 벡터공간 관점:

상태공간: 벡터공간

```
y = [y₁, y₂, ..., yₙ]
```

연산자: 선형변환

```
D(y) = dy/dx
```

방정식:

```
D(y) = Ay

고유벡터 문제!
```

해법:

```
1. A의 고유벡터 v, 고유값 λ 찾기
2. 해: y = e^(λt)v
```

연결:

```
미분방정식 → 선형대수 문제
```

예 2: 푸리에 급수 ↔ 기저 변환

푸리에 급수:

```
f(x) = a₀ + Σ(aₙcos(nx) + bₙsin(nx))
```

추상 벡터공간 관점:

함수공간: $L^2[0, 2\pi]$

기저:

```
{1, cos(x), sin(x), cos(2x), sin(2x), ...}
```

직교성:

```
∫cos(mx)cos(nx)dx = 0  (m ≠ n)
```

좌표:

```
f ↔ [a₀, a₁, b₁, a₂, b₂, ...]
```

연결:

```
푸리에 급수 = 함수를 직교 기저로 표현
            = 선형대수의 기저 변환!
```

예 3: 양자역학 ↔ 힐베르트 공간

양자 상태:

```
|ψ⟩ (벡터)
```

관측:

```
Â (선형 연산자)
```

측정값:

```
고유값
```

확률:

```
|⟨φ|ψ⟩|²  (내적)
```

연결:

```
양자역학 = 무한차원 벡터공간의 선형대수
```

예 4: 데이터 분석 ↔ 차원 축소

데이터:

```
고차원 벡터
```

PCA:

```
공분산 행렬의 고유벡터
= 주성분
```

차원 축소:

```
고유값 큰 방향만 유지
= 부분공간으로 투영
```

연결:

```
통계 = 벡터공간의 기하학
```

(c) 함수공간에 선형대수 적용 가능한 이유:

공리적 관점:

핵심:

```
선형대수 이론 = 8개 공리에서만 도출
```

과정:

Step 1: 공리 확인

함수공간 F:

```
(f + g)(x) = f(x) + g(x)
(c·f)(x) = c·f(x)
```

8개 공리:

```
A1: f + g = g + f ✓
A2: (f + g) + h = f + (g + h) ✓
A3: 0(x) = 0 존재 ✓
A4: (−f)(x) = −f(x) 존재 ✓
M1: c(f + g) = cf + cg ✓
M2: (a + b)f = af + bf ✓
M3: a(bf) = (ab)f ✓
M4: 1·f = f ✓
```

모두 만족!

Step 2: 벡터공간 확정

```
F는 벡터공간 ✓
```

Step 3: 정리 적용

선형대수의 정리:

```
"V가 벡터공간이면, ..."
```

F에 자동 적용:

```
F가 벡터공간이므로, ...
```

예:

정리: "기저는 선형독립이고 span한다"

```
화살표 ✓
함수 ✓ (자동!)
```

정리: "선형변환 T는 기저의 상으로 결정"

```
행렬 ✓
미분 연산자 ✓ (자동!)
```

구체적 예:

선형독립:

정의 (추상):

```
c₁v₁ + ... + cₙvₙ = 0 → cᵢ = 0
```

화살표:

```
c₁→v₁ + ... + cₙ→vₙ = →0 → cᵢ = 0
```

함수:

```
c₁f₁ + ... + cₙfₙ = 0(x) → cᵢ = 0
```

같은 정의, 다른 대상!

기저:

정의 (추상):

```
span하고 선형독립인 집합
```

화살표 ($\mathbb{R}^3$):

```
{î, ĵ, k̂}
```

함수 (P_2):

```
{1, x, x²}
```

같은 개념, 다른 공간!

선형변환:

정의 (추상):

```
T(v + w) = T(v) + T(w)
T(cv) = c·T(v)
```

행렬 ($\mathbb{R}^n \to \mathbb{R}^m$):

```
Ax
```

미분 (함수 → 함수):

```
d/dx
```

같은 성질, 다른 구현!

결론:

함수공간에 선형대수 적용 가능한 이유:

1. **공리 만족** → 벡터공간
2. **이론은 공리만 사용** → 자동 적용
3. **구체적 실현 무관** → 보편성

핵심:

```
추상화 = 공통 구조 추출
→ 한 번 증명 = 모든 곳 적용
```

이것이 수학의 힘입니다!

```
Dataview (inline field 'a·1 + b·(1+x) + c·x²
= (a+b) + bx + cx²

a+b = 2
b = 3
c = 5

→ a = -1, b = 3, c = 5'): Error:
-- PARSING FAILED --------------------------------------------------

> 1 | a·1 + b·(1+x) + c·x²
    |  ^
  2 | = (a+b) + bx + cx²
  3 |

Expected one of the following:
```

```
'(', '*' or '/' or '%', '+' or '-', '.', '>=' or '<=' or '!=' or '=' or
'>' or '<', '[', 'and' or 'or', /[0-9\p{Letter}_-]/u, EOF, text
```