

### 학습목표

1. 차이점과 공통점을 활용하여 내용을 정리할 수 있다.
2. 기술 지문에서 용어를 정리하며 지문을 읽을 수 있다.
3. '~할 수 있다'라는 표현과 문제 선지 구성 간의 관계를 확인할 수 있다.
4. 기술 지문에서 한계와 대응책을 통한 지문 전개를 확인할 수 있다.

# Theme 6.

## 기술 지문 해석과 빠른 문제 해결

출처: 2023학년도 10월 모평 시험 14~17번 문제

# 해설자료 구성과 활용 하는 방법

## [해설 자료]

### 1. 지문

[1 문단]

제도의 선택에 대한 설명에는 합리적인 주체인 사회  
 성원들이 사회 전체적으로 가장 이익이 되는 제도를  
 취한다고 보는 효율성 시각과 이데올로기·경로의존성·  
 역 과정 등으로 인해 효율적 제도의 선택이 일반적  
 않다고 보는 시각이 있다.

제도를 선택하는 것에 대한 설명하는 관점은 두 가  
 있다.

- 효율성 시각
- 효율적 제도의 선택이 일반적이지 않다고 보는 시각

**설명:**

- 두꺼운 두 줄로 사이에 있는 내용은 지문입니다.
- 지문 아래에 해당 문장에 대한 해설을 작성했습니다.

**To-Do List:**

- 지문 읽으시고, 아래 해설을 확인하시면 됩니다.
- 해당 지문이 이해가 안가거나 인상적인 내용이 있다면, 별표나 나름대로의 표시하고, 나중에 복습하시면 됩니다.

### 2. 단어 확인

**WORD 1** 직관-적 直觀的 이라는 것

直 곧을 직 觀 불 관 的 과녁 적

1. 판단이나 추리 따위의 사유 작용을 거쳐 지  
 상을 직접적으로 파악하는 것.

**WORD 2** 호소-력 呼訴力 이라는 것

呼 부를 호 訴 호소할 소 力 힘 력(역)

1. 강한 인상을 주어 마음을 사로잡을 수 있는  
 2. 호소(呼訴)하여 동정(同情)을 불러일으키는

**설명:**

- 평소에 쉽게 지나가는 단어들, 그런 단어들의 뜻을 직접 확인할 수 있도록 구성되었습니다.

**To-Do List:**

- 해당 단어의 뜻을 명확히 몰랐다면, 꼭 뜻을 확인합시다.
- 나중에 공부하면서 지나가는 단어 중에 모르는 단어가 있다면, 나름대로 검색해서 확인해주시길 바랍니다.

### 3. 추가적인 학습 포인트

**CHECK 2** OR이 있는 문장구조(복문의 함정)

OR구조로 이어질 때 문장 성분이 생략되는 경우

- 사과가 바닥으로 떨어지는 이유는 중력으로 설명할
- 포탄이 포물선 운동을 하는 이유는 중력으로 설명할 (OR을 통한 문장구조)

사과가 바닥으로 떨어지거나 포탄이 포물선 운동을 하  
 중력으로 설명 할 수 있다.

해당 네모를 친 표현에서 실제로 생략되는 문  
 '지는 이유는 중력으로 설명할 수 있다.'이며, 이

**설명:**

- 주어진 문장에서 추가로 학습할 수 있는 포인트들을 정리했습니다.

**To-Do List:**

- 학습 포인트를 읽고, 지문에서 해당 포인트를 직접 확인합시다.
- 복습하신다면, 꼭 별표나 자신만의 표시를 합시다.

### 4. 학습목표

학습목표

- 익숙해서 쉽게 지나가는 단어를 제대로 모르는 경우
- 익숙한 문장구조에서 생략되는 표현이 있다는 것을
- 지문에서 그래프의 내용을 확인하고 그래프의 특징
- 개념 간의 차이를 인식하고 구분 기준을 확인할 수
- 정성적 개념과 정량적 개념이 무엇인지 알 수 있다.

**설명:**

- 처음에 정리된 학습목표들은 다시 확인하도록 구성되었습니다.

**To-Do List:**

- 학습목표를 확인하시면서 앞에 네모 칸에 체크합시다
- 학습목표 아래에 네모 칸에, 해당 학습을 하면서 기억에 남거나, 깨달은 점들을 정리합시다.

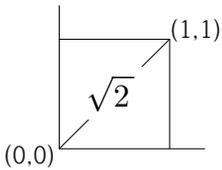
[1문단]

일반적으로 거리는 두 개의 지점이 공간적으로 **떨어진 정도**를 나타내는 물리적 개념이다. 2차원 평면에 두 지점이 (0, 0)과 (1, 1)에 있다면 두 지점 사이의 최단 거리는 두 점을 잇는 직선의 길이  $\sqrt{2}$ 가 된다. 한편 거리는 추상적인 성질이나 가치에 대한 차이를 나타내는 척도로도 사용될 수 있다.

물리적 거리

: 두 개의 지점이 공간적으로 떨어진 정도

예시가 제시됨



추상적 거리

: 추상적인 성질이나 가치에 대한 차이를 나타내는 척도

WORD 1 척도 尺 丈 척 度 법도 도

1. 자료 재는 길이의 표준.
2. 평가하거나 측정할 때 의거할 기준.



다시 말해, 추상적이더라도 하더라도 '거리'라는 것은 두 개념 사이에 얼마나 다른지를 확인할 수 있는 기준이나 표준이라는 것이다.

이럴 경우 떨어진 정도를 나타내는 기능은 유지되지만, 기준이나 관점에 따라 거리를 계산하는 방법이 달라진다.

물리적 거리가 두 개의 지점이 공간적으로 떨어진 정도를 나타내는 기능을 가진 것처럼,

추상적 거리에서도 두 개념이 얼마나 떨어져 있는지 나타내는 기능을 한다.

내용적으로 위에서 추상적 거리에 대한 정의가 반복되는 것처럼 보이지만, 이 문장에서는 물리적 거리와 추상적 거리의 기능적 공통점을 말한 것이다.

한편 개념의 종류나 개념을 인식하는 방식에 따라 거리를 계산하는 방법이 달라진다고 말하며 1문단이 마무리 된다.

여기까지 읽었으면, 이후 추상적 개념의 거리의 예시와 해당 거리들을 계산하는 방법에 대해서 나올 수도 있다는 예측을 할 수 있다.

CHECK 1 차이점과 공통점을 확인해야 한다.

기술 지문 뿐만 아니라, 모든 지문에서 개념 간의 차이점과 공통점은 항상 존재한다.

개념 간의 공통점이 없다면, 다른 두 개념이 연결되는 부분이 없어서 독자에게 맥락이 끊기는 느낌을 주며 독서에 의한 피로감을 증가시킨다.

개념 간의 차이점이 없다면, 동일 개념이 반복된다는 측면에서 안정감을 줄 수 있지만, 독자에게 지루함 혹은 '왜 이것을 계속 읽어야 하는지'에 대한 의문을 주게 된다.

따라서 개념간의 차이점과 공통점을 적절히 사용하며 적절한 개념의 변화를 추구하며, 전달하고자 하는 정보를 정리한다. 다시 말해, 정보의 변화를 정확히 캐치하기 위해 개념 간의 공통점과 차이점을 확인하며 읽어야 한다.

기술 지문은 기술이 실제로 적용되어 어떻게 의미있는 작용으로 이어지는 지를 정리한다. 따라서 원리를 정의하는 것으로부터 제품이 움직이는데 기술이 어떻게 사용되는 지까지의 과정을 파악해야 한다. 과정을 파악하기 위해서는 개념의 변화를 파악해야 한다. 다시 말해, 기술 지문에서도 꼭 차이점과 공통점을 찾아야 한다는 것이다.

## [2문단]

거리의 개념은 디지털 데이터에도 적용될 수 있다. 데이터 간의 거리는 추상적 거리의 개념으로, 데이터가 표현하려는 정보에 따라 측정 방법이 다르다.

(추상적인) 경우 떨어진 정도를 나타내는 기능은 유지되지만, 기준이나 관점에 따라 거리를 계산하는 방법이 달라진다. 다시 말해, 1문단에서 정리된 내용을 디지털 데이터에 적용한 것이다.

00, 11과 같은 2비트의 데이터가 2진수로 표현된 수치를 가리킨다면 00과 11의 거리는 두 수치의 차이  $|(0 \times 2^0 + 0 \times 2^1) - (1 \times 2^0 + 1 \times 2^1)| = 3$ 이 된다. 그런데 2비트의 데이터 00이나 11이 어떤 상태를 나타내는 부호라면 거리는 두 부호가 구별되는 정도라 할 수 있다.

2비트의 데이터 = 2진수로 표현된 수치

- 부호: 어떤 상태를 나타냄

예시) 2비트의 데이터 00이나 11

- 거리: 두 부호가 구별되는 정도

예시) 두 수치의 차이

$$|(0 \times 2^0 + 0 \times 2^1) - (1 \times 2^0 + 1 \times 2^1)| = 3$$

부호와 거리가 다르다는 점을 기억하자!

해밍 거리는 부호의 관점에서 부호들 간의 거리를 표현하는 방법 중 하나이다. 해밍 거리는 길이가 같은 두 부호를 비교하였을 때 두 부호의 같은 자리에 있는 서로 다른 문자의 개수로 나타낸다. 예를 들어 세 개의 부호 00, 01, 11이 있다면 00과 01의 해밍 거리는 1이고, 00과 11의 해밍 거리는 2이다.

해밍 거리는 2진수로 표현된 수치와 부호는 동일할 수 있지만, 길이를 구하는 방식이 달라지는 경우다.

해밍 거리:

두 부호의 같은 자리에 있는 서로 다른 문자의 개수

예시 1) 00과 01의 해밍 거리는 1

예시 2) 00과 11의 해밍 거리는 2

이때 부호들 간의 최소 해밍 거리는 1이고, 최대 해밍 거리는 2이다.

최소 해밍 거리는 ‘해밍 거리의 최소값’이고, 최대 해밍 거리는 ‘해밍 거리의 최대값’이다.

- 해밍 거리의 최소값: 1

- 해밍 거리의 최대값: 2

‘예시 1) 00과 01의 해밍 거리는 1’은 해밍 거리의 최소값의 예시이며,

‘예시 2) 00과 11의 해밍 거리는 2’는 해밍거리의 최대값의 예시이다.

**CHECK 2** 기술 지문에서 용어를 정리하는 것

“물리적 거리”

“추상적 거리”

1문단에서 정리된 거리는 개념이 디지털로 확장되고, 이후 2비트의 데이터로 더 구체적인 예시가 나왔다.

“부호”

“거리”

해당 용어들을 정리하면서 데이터가 ‘2진수로 표현될 때’와 ‘해밍거리로 표현될 때’를 구분할 수 있었다. ‘부호’는 동일하지만, ‘거리’를 구하는 방식이 달라진다.

한편 생소한 개념인 ‘해밍거리’를 별다른 불편함 없이 받아들일 수 있었다.

“해밍 거리의 최소값”

“해밍 거리의 최대값”

해밍 거리의 ‘최소값’과 ‘최대값’으로 개념이 구분된다. 분명 해당 개념의 구분을 통해 또 다른 개념으로 이어질 것이다.

이처럼 정리된 용어는 이후 용어를 이해하기 위한 밑바탕이 된다. 다시 말해, 기술 지문을 막힘없이 정리해 나가기 위해서는 반드시 용어를 정리해야 한다.

### [3문단]

부호들 간의 최소 해밍 거리를 충분히 멀게 한다면 통신이나 저장 과정에서 발생하는 오류를 검출하여 수정할 수 있다.

- ① 부호들 간의 최소 해밍 거리를 충분히 멀게 한다  
= 해밍 거리의 최소값이 충분히 크다.
- ②-1. 통신이나 저장 과정에서 발생하는 오류를 검출할 수 있다.
- ②-2. 통신이나 저장 과정에서 발생하는 오류를 수정할 수 있다.

이후 주어진 예시에서는, 해밍 거리의 최소값이 충분히 클 때 통신이나 저장과정에서 발생하는 오류를 확인하고 수정하는 과정에 대해 정리될 것이라는 예상할 수 있다.

한편 부호를 통해 통신하거나 저장을 할 수 있다는 점을 알 수 있다. 이는 2문단에서 '디지털 데이터에도 적용되'는 점과 연결하여 생각할 수 있다.

예를 들어 전송하려는 1비트의 원시 부호 0과 1이 있고 부호 단위로 송수신한다고 가정해 보자. 송신자가 1을 보낸다면 수신자는 0이나 1 중 하나를 받게 될 것이고, 송신자가 어떤 데이터를 보냈는지 알 수 없기 때문에 오류가 발생하더라도 오류가 있는지 알 수 없다.

1비트의 원시 부호 = 한 자리수 부호

: 0 또는 1

송신자 보낸 데이터: 1

수신자가 받는 데이터: 0 또는 1

이 경우에는 통신이나 저장 과정에서 발생하는 오류를 검출할 수도 없으며 수정할 수도 없다. 따라서 지금의 예시는 적절한 예시가 아니다.

이 경우 부호들 간의 최소 해밍 거리는 1이다.

한 자리수 부호에서 최소 해밍 거리는 1이며, 이는 부호들 간의 최소 해밍 거리를 충분히 멀지 못한 경우이다. 따라서 다른 자리수의 부호 예시가 제시되어야 한다.

0이나 1을 송수신하는 대신 원시 부호(x) 뒤에 확인 부호(p)를 덧붙여 x p에 해당하는 2비트 단위의 전송 부호를 만들어 보자.

2비트 데이터 = 두 자리수 부호

= 앞은 원시 부호(x) / 뒤는 확인 부호(p)

㉠ 전송 부호는 고정된 원시 부호에 확인 부호를 덧붙이고, 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다는 규칙을 정한다면 전송 부호는 0 0과 1 1이 된다.

추가된 규칙: 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다.

- 0 0일 경우는 1의 개수가 0,

- 1 1일 경우는 1의 개수가 2

즉, 두 가지 경우 모두 추가된 규칙을 준수한다.

만일 수신자가 0 1이나 1 0 중 하나를 받은 경우 전송 부호에 오류가 있음을 알 수 있다. 하지만 어느 자리에서 오류가 났는지 알 수 없기 때문에 오류를 수정할 수는 없다.

- 0 1이나 1 0 중 하나를 받은 경우는 1의 개수가 1  
추가된 규칙을 준수하지 못한다. 이를 통해 전송 부호에 오류가 검출하는 것은 가능했지만, 오류가 어느 자리에서 났는지를 알 수가 없다. 따라서 오류를 수정할 수가 없다.

다시 말해, 아직 온전한 예시를 제시되지 않았다는 것이며, 이후 통신이나 저장 과정에서 발생하는 오류를 수정할 수 있는 경우에 예시가 될 것이라는 예상을 할 수 있다.

[4문단], 그리고 [A]

0 0이나 1 1을 송수신하는 대신 p와 동일한 규칙의 확인 부호(q)를 한 번 더 덧붙여 x p q에 해당하는 3비트 단위의 전송 부호 0 0 0과 1 1 1 중 하나를 송수신한다고 가정해 보자.

그래서 이번에 제시된 예시는  
3비트의 데이터 = 세 자리 수 부호  
= 앞은 원시 부호(x)  
/ 가운데는 확인 부호(p) / 마지막은 확인부호(q)

또한 동일한 규칙이 제시되기 때문에, 추가된 규칙인 '확인 부호는 원시 부호에 대한 1의 짝수가 되도록 만들어야 한다'는 점을 기억해야 한다.

따라서 전송부호는 두 가지, 0 0 0과 1 1 1이다.

한 자리의 오류만 있다고 가정하면 수신자가 0 0 1, 0 1 0, 1 0 0, 0 1 1, 1 0 1, 1 1 0 중 하나를 받은 경우 오류 발생 자리를 검출하여 수정할 수 있다.

한 자리만 오류가 있다고 가정하면  
수신자가 받을 수 있는 부호  
: 0 0 1, 0 1 0, 1 0 0, 0 1 1, 1 0 1, 1 1 0

예를 들어 1 1 0의 경우 x인 1에 대해 p와 q는 각각 1이 되어야 1의 개수가 짝수가 되지만 q가 0이므로 1의 개수가 홀수이다. 따라서 오류 발생 자리를 검출하여 1 1 0을 1 1 1로 수정할 수 있다.

수신자가 받은 부호를, 1 1 0 하나로 고정함: 확인 부호와 원시부호 사이의 규칙을 고려하여 상황을 정리하면,  
- x와 p에서의 1의 개수는 2  
- x와 q에서의 1의 개수는 1  
x와 p에서의 1의 개수 규칙은 준수되고 있지만, x와 q에서의 1의 개수 규칙은 준수되지 않는다. 따라서 x와 p의 경우는 수정이 필요하지 않으며, x와 q의 경우에는 수정이 필요하다.

따라서 수정되어야 하는 부호는 q이며, 이 과정을 통해 1 1 1로 수정되는 것을 확인할 수 있다.

이 경우 전송 부호 간의 최소 해밍 거리가 3이어서 한 자리의 오류를 검출하여 수정할 수 있는 것이다.

전송부호는 두 가지, 0 0 0과 1 1 1이다.  
이 경우 부호에서 최소 해밍 거리는 3이다. 이번에 제시된 예시는 두 가지를 만족한다.

- ②-1. 통신이나 저장 과정에서 발생하는 오류를 검출할 수 있다.
- ②-2. 통신이나 저장 과정에서 발생하는 오류를 수정할 수 있다.

이는 송신부호들 간의 최소 해밍 거리를 충분히 먼 경우라고 해석할 수 있다.

CHECK 3 '~할 수 있다'의 중의성

'-(으)ㄹ 수 있다' 구문은 가능성 혹은 능력의 의미를 나타냅니다. 예를 들면 '나는 너에게 이 물건을 팔 수 있다.'라고 하면 '물건을 팔 가능성(능력)이 있다.'라는 것 그 자체를 의미합니다. '-(으)ㄹ 수 있다' 구문이 애매모호하게 해석된다고 느껴진다면 그것은 '가능성'이라는 것 자체가 '앞으로 실현될 수 있는 성질.'을 뜻하는 것이기 때문입니다. 따라서 '팔 수 있다'라고 하면 '이 물건을 팔아야 되는지 또는 안 팔아도 되는지' 하는 것과는 관련이 없다 할 수 있습니다.

만약 '~할 수 있다'가 가능성의 의미로 활용되었다면, 3비트의 데이터가 전송되었다고 하더라도 오류가 수정되지 않는 경우가 있다고 해석할 수 있다.

기술 지문은 경우의 수를 묻는 문제를 구성하는 경우가 많다. 문제의 선지 중에 난이도가 높은 선지는 생각하기 힘든 경우로 설정되는 경우가 많다.

'~할 수 있다'는 표현에서 난이도가 높은 선지가 구성되는 것은 15번 문제에서 확인 할 수 있다.

CHECK 4 기술적 한계와 확장

1비트의 원시 부호

이 경우에는 통신이나 저장 과정에서 발생하는 오류를 검출할 수도 없으며 수정할 수도 없다. 다시 말해, 오류 자체에 대해 아무런 것도 할 수 없는 기술적 한계가 제시되었다.

2비트: 원시 부호  $x$  와 확인 부호  $p$

+ 규칙이 추가됨: 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다.

추가된 규칙을 준수하지 못한다. 이를 통해 전송 부호에 오류가 검출하는 것은 가능했지만, 오류가 어느 자리에서 났는지를 알 수가 없다. 따라서 오류를 수정할 수가 없다. 다시 말해, 아직 오류를 수정하지 못하는 기술적 한계가 제시되었다.

3비트: 원시 부호  $x$ 와 확인 부호  $p, q$

+ 규칙이 추가됨: 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다.

이 경우 부호에서 최소 해밍 거리는 3이다. 이번에 제시된 예시는 두 가지를 만족한다.

②-1. 통신이나 저장 과정에서 발생하는 오류를 검출할 수 있다.

②-2. 통신이나 저장 과정에서 발생하는 오류를 수정할 수 있다.

이는 송신부호들 간의 최소 해밍 거리를 충분히 먼 경우라고 해석할 수 있다. 다시 말해, 기술적 한계에 대한 대응책이 마련되었다고 해석할 수 있다.

기술적 한계를 제시하면서 기술의 내용을 확장해나가는 구성은 자주 반복된다. 이런 경우 헛갈리지 않고 내용을 정리할 수 있는 방법은 기술적 한계를 확인하는 것이다. 한계를 확인하고 이에 대한 대응 방법이 어떤 방식으로 마련되는지를 정리할 수 있어야 한다.

[5문단]

원시 부호에 확인 부호를 충분히 덧붙이면 전송 부호의 길이는 길어지지만 전송 부호들 간의 최소 해밍 거리도 함께 멀어져 오류가 많이 발생하더라도 오류를 검출하여 정하는 것이 가능하다.

① 확인 부호를 충분히 덧붙이면,

②-1. 전송 부호의 길이가 증가함

②-2. 전송 부호들 간의 최소 해밍 거리도 함께 멀어짐

③-1. 오류가 많이 발생함

③-2. 오류를 검출하여 수정하는 것이 가능함

‘③-2.’는 확인부호를 덧붙였을 때의 장점으로 해석할 수 있다.

하지만 동일한 정보를 보낼 때 덧붙이는 확인 부호의 개수가 늘어나면 보내야 하는 데이터의 양이 늘어나 전송 효율이 낮아진다

(① 확인 부호를 충분히 덧붙이면,)

②-1. 데이터의 양이 늘어남

②-2. 전송효율이 낮아진다.

‘②-2.’는 확인부호를 덧붙였을 때의 단점으로 해석할 수 있다

14.④ 15.④ 16.① 17.①

[14번 문제]

윗글을 통해 알 수 있는 내용으로 적절하지 않은 것은?

추론문제다.

윗 글을 통해 엄밀하게 정의할 수는 없지만,

- 추론이 가능한 선지가 있을 것이고,
- 혹은 아예 추론이 가능하지 않은 선지가 있을 것이다.

해당 발문에서는 적절하지 않은 것을 찾는 것이기 때문에, 선지 중에서 추론이 가능하지 않은 선지를 찾아야 한다.

① 2진수로 표현된 수치를 가리키는 데이터들 간의 거리는 수치 간의 차로 표현될 수 있다.

2문단에서, '00, 11과 같은 2비트의 데이터가 2진수로 표현된 수치를 가리킨다면 00과 11의 거리는 두 수치의 차이인  $|(0 \times 2^0 + 0 \times 2^1) - (1 \times 2^0 + 1 \times 2^1)| = 3$ 이 된다'는 것을 확인 할 수 있다. 다시 말해서, 2진수로 표현된 수치를 가리키는 데이터들 간의 거리는 수치 간의 차를 통해 표현할 수 있다는 말이며, 따라서 해당 선지는 추론이 가능하다.

② 추상적인 성질이나 가치의 차이를 나타내는 척도로 거리의 개념이 사용될 수 있다.

1문단에서, '거리는 추상적인 성질이나 가치에 대한 차이를 나타내는 척도로도 사용될 수 있'기 때문에, 따라서 해당 선지는 추론이 가능하다.

③ 물리적 개념에서의 거리는 두 지점이 공간적으로 떨어져 있는 정도를 나타낸다.

1문단에서, '일반적으로 거리는 두 개의 지점이 공간적으로 떨어진 정도를 나타내는 물리적 개념'이다. 따라서 해당 선지는 추론이 가능하다.

④ 00과 11의 2진수 수치의 차이와 해밍 거리는 같은 값으로 측정된다.

1번 선지에서, 00과 11의 2진수 수치의 차이가 3임을 알 수 있다.

2문단에서, 00과 11의 해밍 거리는 2이다. 따라서 해당 선지는 추론이 불가능하다.

⑤ 데이터가 표현하려는 정보에 따라 거리를 측정하는 방법이 다르다.

2문단에서, '데이터가 표현하려는 정보에 따라 측정 방법이 다르다'는 것을 알 수 있으며, 4번 선지와 연결해서 생각해보면 동일 부호라고 하더라도 2진수 수치상의 거리를 구하는 방법과 해밍 거리를 구하는 방법이 다르다는 점도 생각할 수 있다. 어찌되었든, 해당 선지는 추론이 가능하다.

[15번 문제]

[A]와 <보기>를 이해한 내용으로 적절하지 않은 것은?  
[3점]

[A]는 4문단이며, 3비트의 데이터를 통해 원시 부호 x와 확인 부호 p, q가 제시된다.

또한 기존 규칙, 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다는 경우도 있다는 점을 기억해야 한다.

이 경우 부호에서 최소 해밍 거리는 3이다. 이번에 제시된 예시는 두 가지를 만족한다.

- ②-1. 통신이나 저장 과정에서 발생하는 오류를 검출할 수 있다.
- ②-2. 통신이나 저장 과정에서 발생하는 오류를 수정할 수 있다.

이는 송신부호들 간의 최소 해밍 거리를 충분히 먼 경우라고 해석할 수 있다.

< 보기 >

확인 부호가 오류 발생 자리에 대한 정보가 되도록 규칙을 정하면 전송 부호에서 한 자리 오류가 발생했을 때 수정이 가능하다. 확인 부호를 검사하여 p에 오류가 있으면 **p 자리**를 1로, 오류가 없으면 0으로 표현한다. 같은 방식으로 q에 오류가 있으면 **q 자리**를 1로, 오류가 없으면 0으로 표현한다. 0과 1로 표현된 **p 자리** **q 자리**를 계산하면 한 자리의 오류가 발생했을 때 그 자리를 알아낼 수 있다.

송신	수신	규칙		오류 발생 자리	
		오류			계산
		p 자리	q 자리		
000	000	0	0	$0 \times 2^1 + 0 \times 2^0$	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	010		0	$1 \times 2^1 + 0 \times 2^0$	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
	110	0	1	$0 \times 2^1 + 1 \times 2^0$	
	011	1	1	$1 \times 2^1 + 1 \times 2^0$	
⋮	⋮	⋮	⋮	⋮	⋮

<보기>도 3비트의 데이터에 대한 분석 자료이며, 따라서 [A]에서 정리된 내용을 기억하며 문제에 접근해야 한다.

① 송신자는 전송 부호 간의 해밍 거리가 3이 될 수 있도록 0은 0 0 0으로, 1은 1 1 1로 보내는 것이겠군.

4문단에서, ‘3비트 단위의 전송 부호 0 0 0과 1 1 1 중 하나를 송수신한다고 가정’했다. 이 때, 최소 해밍거리는 3이 된다. 따라서 해당 선지는 추론이 가능하다.

② 수신자가 0 1 0을 받았다면 p 자리의 오류를 1로 표현하여 0 0 0으로 판단하겠군.

기존 규칙, 확인 부호는 원시 부호에 대한 1의 개수가 짝수가 되도록 만든다는 것을 ‘짝수규칙’이라 하겠다.

- x와 q의 1의 개수는 0, 짝수규칙을 만족한다.
- x와 p의 1의 개수는 1, 짝수규칙을 만족하지 않는다.

x와 q의 관계에서 수정할 필요가 없지만, x와 p의 관계는 수정해야만 한다. x는 수정할 필요가 없는 관계에 속해있기 때문에, 남은 p는 1에서 0으로 수정해야 한다. 따라서 해당 선지는 추론이 가능하다.

③ 수신자가 1 1 0이나 1 0 1을 받았다면 수신한 부호에 있는 0을 1로 수정하여 모두 1 1 1로 판단하겠군.

- 110의 경우는 q를 0에서 1로 수정해야 하며
  - 101의 경우는 p를 0에서 1로 수정해야 한다.
- 결국 모두 1 1 1로 판단할 수밖에 없다. 따라서 해당 선지는 추론이 가능하다.

---

④ 수신자가 011을 받았다면 p 자리 와 q 자리 모두에 오류가 있는 경우이므로 두 자리의 오류를 수정하겠군.

---

011의 경우, x와 q의 1의 개수 및 x와 p의 1의 개수가 1이며, 모두 짝수규칙을 만족하지 않는다.  
따라서 수정해야 하는 방향이 두 가지로 정리할 수 있다.  
- x를 0에서 1로 수정한다.  
- p와 q를 0으로 수정한다.  
결과값은 1 1 1 혹은 0 0 0일 것이다.

물론 <보기>의 표의 네 번째 칸에 거리 계산을 통해 0 0 0이 0 1 1로 수신된 상황을 확인 할 수 있지만, 이후 몇 번째가 될지 모르는 칸에서 1 1 1이 0 1 1로 수신될 상황 역시 표현될 것이다.

따라서 해당 <보기>는 함정을 유도하기 위한 도구로 고안된 것이라 판단할 수 있다.

또 다른 측면에서, 4번째 문단에 대한 해설을 하면서, '할 수 있다'에 대한 해석을 말했다.  
다시 말해, 0 1 1 이나 1 0 0의 경우에는 전송 부호에 오류가 검출하는 것은 가능했지만, 오류가 어느 자리에서 났는지를 알 수가 없다. 따라서 오류를 수정할 수가 없다.  
결국 성공적으로 예외가 수정되었음에도, 결국 이런 예외적인 상황이 존재하기 때문에, 4문단에서 '~할 수 있다'라는 표현을 사용한 것이다.

각설하고, 4번 선지는 거짓이다.

---

⑤ 수신자가 1 1 1을 받았다면 p자리와 q자리의 오류를 모두 0으로 표현하여 오류가 없는 것으로 판단하겠군.

---

수신자가 1 1 1을 받았다면, x와 q의 1의 개수 및 x와 p의 1의 개수가 2이며, 모두 짝수규칙을 만족한다.  
따라서 해당 선지는 추론 가능하다.

## [16번 문제]

---

①에 대한 이해로 가장 적절한 것은?

---

① 전송 부호는 고정된 원시 부호에 확인 부호를 덧붙이고 즉, 비트 수가 커지는 상황에 대해 물어보고 있다.

---

① 전송 부호들 간의 최소 해밍 거리를 멀게 하면 전송하는 데이터의 양은 늘어난다.

---

5문단에서 '동일한 정보를 보낼 때 덧붙이는 확인 부호의 개수가 늘어나면 보내야 하는 데이터의 양이 늘어'난다. 따라서 해당 선지는 추론가능하다.

---

② 전송 부호들 간의 최소 해밍 거리가 1이면 전송 과정에서의 오류 검출이 가능하다.

---

1비트 데이터의 경우에는 통신이나 저장 과정에서 발생하는 오류를 검출할 수도 없으며 수정할 수도 없다. 이는 3문단에서 확인할 수 있으며, 해당 선지는 거짓이다.

---

③ 두 전송 부호의 같은 자리에 같은 문자의 개수가 많을수록 해밍 거리는 멀어진다.

---

'해밍 거리는 길이가 같은 두 부호를 비교하였을 때 두 부호의 같은 자리에 있는 서로 다른 문자의 개수로 나타'내기 때문에, 두 전송 부호의 같은 자리에 같은 문자의 개수가 많을수록 해밍거리는 가까워진다. 따라서 해당 선지는 거짓이다.

---

④ 덧붙이는 확인 부호가 많아지면 전송 부호들 간의 최대 해밍 거리는 가까워진다.

---

3번 선지에서, 해밍 거리의 정의를 잊지말자!

확인 부호가 많아지면, 부호의 자릿수는 증가한다. 따라서 해밍 거리의 최대값은 증가한다. 따라서 최대 해밍 거리는 가까워지지 않으며, 해당 선지는 거짓이다.

---

⑤ 전송 부호들 간의 최소 해밍 거리가 가까워질수록 전송 효율은 낮아진다.

---

‘확인 부호의 개수가 늘어나면 보내야 하는 데이터의 양이 늘어나 전송 효율이 낮아’지는 것이며, 확인 부호의 개수가 증가하면서, 최소 해밍 거리는 멀어진다.

최소 해밍 거리와 확인 부호의 개수 간의 관계를 고려하여 5번 선지를 분석하면, 최소 해밍 거리가 가까워질수록 전송 효율은 높아질 수 있다고 예측할 수 있다. 따라서 해당 선지는 거짓이다.

[17번 문제]

---

㉔의 문맥적 의미와 가장 유사한 것은?

---

일반적으로 거리는 두 개의 지점이 공간적으로 ㉔떨어진 정도를 나타내는 물리적 개념이다.

즉, ㉔의 문맥적 의미와 연관된 논리 개념은 공간이다.

---

① 식당은 본관과 조금 떨어져 있는 별관이다.

---

: 1번은 식당과 본관이 공간적으로 거리가 있음을 말하고 있다.

---

② 해가 떨어지자 새는 보금자리로 돌아갔다.

---

: 해가 떨어지는 것은 시간적으로 저녁 혹은 밤이 되었음을 말한다.

---

③ 그들의 실력은 평균보다 떨어지는 편이다.

---

: 실력은 추상적인 개념이며, 개인들의 능력에 차이가 있음을 말한다.

---

④ 상처가 나서 생긴 딱지가 아물어 떨어졌다.

---

: 딱지가 아물어서 떨어지는 것은 아래로 떨어지는 것을 말하며, 이 때 방향성이 있음을 알 수 있다.

---

⑤ 물건을 팔면 본전을 빼고 만 원이 떨어진다.

---

: 본전을 제외하고, 이익이 만원이 된다는 것을 말한다. 즉, 회계에서 영업이익에 해당하는 개념이다.

공간의 개념을 통해, 1, 4번으로 추려진다.

다만, ㉔의 경우 방향성은 존재하지 않기 때문에, 결국 4번은 배제가 되며, 남은 1번이 정답이 된다.

