

1. 과자 (4점)

1번부터 7번까지 번호가 붙은 7명의 사람이 있고, 과자가 83개 있다.

과자는 라운드별로 다음과 같이 나누어 주며, 각 라운드 안에서는 1번 사람부터 7번 사람까지 번호 순서대로 나누어 준다.

- 첫 번째 라운드에서는 1번 사람에게 1개, 2번 사람에게 2개, ..., 7번 사람에게 7개를 준다.
- 과자가 남아 있으면 두 번째 라운드에서는 1번 사람에게 7개, 2번 사람에게 6개, ..., 7번 사람에게 1개를 준다.
- 세 번째 이후의 라운드에서는, 홀수 번째 라운드는 첫 번째 라운드와, 짝수 번째 라운드는 두 번째 라운드와 같은 방식으로 진행한다.

지금 차례에서 과자를 받을 사람에게 줄 만큼 과자가 남아 있지 않다면, 그 즉시 나누어 주기를 중단한다.

이때 남아 있는 과자의 개수는?

4

5

6

7

8

2. 수직선과 수평선 (5점)

좌표평면 위에 서로 다른 9개의 점이 있고, 각 점의 좌표는 $(6, 4)$, $(7, 3)$, $(8, 5)$, $(1, 8)$, $(3, 2)$, $(1, 6)$, $(1, 5)$, $(1, 4)$, $(6, 8)$ 이다.

이 점들 가운데 서로 다른 두 점을 지나고 x 축 또는 y 축에 평행한 서로 다른 직선의 개수는?

단, 같은 직선을 결정하는 점 쌍이 여러 개 있더라도 그 직선은 한 번만 센다.

3

4

5

6

7

3. 증가 부분열 (5점)

정수들의 수열 $A = [A_1, A_2, \dots, A_{10}] = [6, 7, 1, 8, 10, 4, 7, 8, 9, 6]$ 이 있다.

A 의 **부분열**은 A 의 원소 가운데 일부를 골라 원래의 순서를 유지하여 얻은 수열을 뜻한다. 앞의 원소보다 뒤의 원소가 항상 큰 부분열을 **증가 부분열**이라고 한다.

예를 들어 $[6, 7, 8]$ 은 A 의 증가 부분열이지만, $[1, 8, 8]$ 은 A 의 부분열이지만 증가 부분열은 아니다.

$1 \leq i \leq 10$ 인 자연수 i 에 대하여, $L(i)$ 를 A 의 i 번째 원소를 첫 원소로 하는 가장 긴 증가 부분열의 길이라고 하자.

$L(1) + L(6)$ 의 값은?

5

6

7

8

9

4. 신발 찾기 (6점)

신발 한 켤레는 왼쪽과 오른쪽 두 짝으로 이루어진다. (즉, '짝'은 신발 한 쪽을 뜻한다.) 25켤레의 신발, 즉 50짝이 있으며, 50짝은 모두 서로 구별된다.

이 50짝 가운데 서로 다른 세 짝을 임의로 고를 때, 고른 세 짝 가운데 같은 켤레에 속한 두 짝이 포함될 확률은?

$\frac{1}{98}$

$\frac{1}{49}$

$\frac{2}{49}$

$\frac{3}{49}$

$\frac{4}{49}$

$\frac{5}{49}$

$\frac{6}{49}$

$\frac{1}{50}$

$\frac{1}{25}$

$\frac{3}{50}$

5. 두 반의 평균 (7점)

10명의 학생의 시험 점수는 각각 8, 17, 9, 7, 4, 3, 18, 17, 5, 16이다.

이 학생들을 6명으로 이루어진 반과 4명으로 이루어진 반으로 나눈다. 각 반에 속한 학생들의 시험 점수의 평균을 각각 A , B 라 할 때, $A + B$ 의 최댓값은?

13

22

23

24

33

6. 1과 2 (8점)

아래 조건을 모두 만족하는 양의 정수 N 가운데 가장 작은 값을 구하라.

- N 의 십진수 표현은 숫자 1과 2로만 이루어져 있다.
- N 의 십진수 표현에서 숫자 1이 정확히 8번 등장한다.
- N 의 십진수 표현에서 숫자 2가 정확히 8번 등장한다.
- N 의 십진수 표현에서 연속한 부분 문자열 '12'가 정확히 4번 등장한다.
- N 의 십진수 표현에서 연속한 부분 문자열 '21'이 정확히 5번 등장한다.

7. 연속한 1이 없는 이진수 (8점)

149의 이진수 표현은 $10010101_{(2)}$ 이며, 이 표현에는 숫자 1이 연속해서 나타나지 않는다.

0 이상 255 이하의 정수 가운데, 이진수 표현에서 숫자 1이 연속해서 나타나지 않는 수의 개수는?

32

34

55

63

89

8. 비트 방정식 (10점)

0 이상의 정수 a, b 에 대하여 $a \oplus b$ 는 a 와 b 의 **비트별 배타적 논리합(XOR)**이다. 즉, a 와 b 의 이진수 표현에서 같은 자리의 두 비트가 서로 다르면 해당 자리에 1을, 같으면 0을 두어 얻는 값이다. (C, C++, Java, Python 언어의 \wedge 연산자와 같다.)

예를 들어,

- $6 \oplus 3$ 의 경우, $6 = 110_{(2)}$, $3 = 011_{(2)}$ 이므로 $6 \oplus 3 = 101_{(2)} = 5$ 이다.
- $10 \oplus 12$ 의 경우, $10 = 1010_{(2)}$, $12 = 1100_{(2)}$ 이므로 $10 \oplus 12 = 0110_{(2)} = 6$ 이다.

양의 정수 n 과 k 에 대하여, $n \star k$ 는 n 을 k 로 나눈 몫을 뜻한다. 예를 들어, $27 \star 4 = 6$ 이다.

다음 식을 만족하는 양의 정수 n 이 유일하게 존재한다. 우변의 값이 이진법으로 표현되었음에 유의하라.

$$n \oplus (n \star 2^2) \oplus (n \star 2^5) = 10011110_{(2)}$$

이 n 을 **이진법으로** 나타내어라.

9. 스트라이크 (10점)

민수와 그의 친구 지수가 다음과 같은 게임을 한다. 민수가 5자리 자연수 하나를 정하면, 지수는 그 수를 맞히기 위해 여러 번 추측한다. 민수는 지수가 추측할 때마다, 숫자가 일치하는 자리의 개수를 알려 준다.

예를 들어, 민수가 정한 수가 12345이고 지수가 10394를 추측했다면, 첫 번째 자리의 1과 세 번째 자리의 3이 일치하므로 알려 주는 개수는 2이다.

지수의 추측과 그 결과가 아래 표와 같을 때, 민수가 정한 5자리 수를 구하라. 정답은 유일하게 결정된다.

| 지수의 추측 | 일치하는 자리의 개수 |
|--------|-------------|
| 40758 | 0 |
| 61409 | 1 |
| 11354 | 2 |
| 86143 | 1 |
| 52752 | 2 |
| 57531 | 1 |
| 58879 | 1 |

10. 꼬리합 (11점)

십진법으로 나타냈을 때 8자리인 양의 정수 N 의 **꼬리합**은 N 의 왼쪽 끝자리부터 0개, 1개, ..., 7개의 숫자를 지워 얻는 8개의 수의 합이다. 숫자를 지운 결과의 앞자리에 생기는 0은 무시하고 수로 본다.

예를 들어, 40891234의 꼬리합은 $40891234 + 0891234 + 891234 + 91234 + 1234 + 234 + 34 + 4 = 42766442$ 이다.

꼬리합이 20260510이면서 십진법으로 나타냈을 때 8자리인 양의 정수 하나를 구하라. 조건을 만족하는 답이 여러 개 존재할 수 있으며, 그중 어느 것을 답해도 정답으로 인정한다.

11. 이진 트리의 거리 (12점)

루트가 있는 이진 트리를 생각하자. 이 트리의 단말 정점은 8개이며, 모든 내부 정점에는 정확히 두 개의 자식이 있다.

단말 정점에는 트리를 중위 순회할 때 방문되는 순서대로 $1, 2, \dots, 8$ 의 번호가 붙어 있다.

두 단말 정점 u, v 사이의 거리 $d(u, v)$ 는 u 에서 v 로 가는 단순 경로에 포함된 간선의 수로 정의한다.

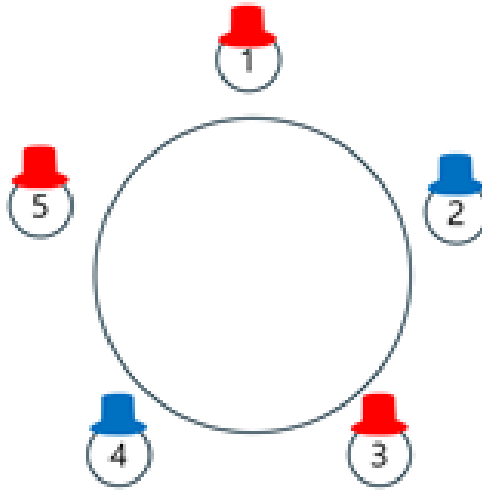
이 트리에서 각 $1 \leq i \leq 7$ 에 대한 $d(i, i+1)$ 의 값은 아래 표와 같다.

| | | | | | | | |
|-------------|---|---|---|---|---|---|---|
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $d(i, i+1)$ | 2 | 6 | 3 | 2 | 4 | 3 | 4 |

위 표의 값을 만족하는 트리는 유일하게 결정된다. $d(1, 5) + d(3, 7) + d(2, 8)$ 의 값을 구하라.

12. 모자 색 맞추기 (12점)

TV 쇼 게임에 참가한 5명의 사람이 아래 그림과 같이 원탁에 둘러앉아 있다. 편의상 각 사람을 1번부터 5번까지의 번호로 구분하자.



게임은 다음과 같이 진행된다.

- 빨간 모자 4개와 파란 모자 3개, 총 7개의 모자 가운데 사회자가 5개를 골라 참가자들에게 하나씩 씌운다.
- 그림과 같이 1번 참가자는 빨간 모자, 2번 참가자는 파란 모자, 3번 참가자는 빨간 모자, 4번 참가자는 파란 모자, 5번 참가자는 빨간 모자를 쓰고 있다.
- 참가자들은 전체 모자 구성이 빨간 모자 4개, 파란 모자 3개라는 사실과, 그중 5개가 자신들에게 씌워졌다는 사실을 안다. 그러나 어떤 5개가 선택되었는지는 알지 못한다.
- 각 참가자는 자기 모자의 색은 볼 수 없지만, 다른 참가자들의 모자 색은 볼 수 있다.
- 사회자는 1번 참가자부터 시작하여 번호 순서대로 “당신은 자신의 모자 색을 알 수 있습니까?”라고 묻는다.
- 질문을 받은 참가자는 자신의 모자 색을 정확히 알 수 있으면 ‘Yes’, 그렇지 않으면 ‘No’라고 답한다.
- 모든 참가자는 이전 참가자들의 답을 모두 들으며, 완벽하게 논리적으로 추론한다.

1번 참가자는 ‘No’라고 답했다.

이때 2번, 3번, 4번 참가자의 답을 순서대로 나열한 것은?

- Ⓐ No, No, No
- Ⓑ No, No, Yes
- Ⓒ No, Yes, No
- Ⓓ No, Yes, Yes
- Ⓔ Yes, No, No

⊗ Yes, No, Yes

⊗ Yes, Yes, No

⊗ Yes, Yes, Yes

13. 앞면과 뒷면 (9점)

9장의 카드가 일렬로 놓여 있다. 각 카드에는 앞면과 뒷면에 각각 수가 하나씩 적혀 있다.

각 카드마다 앞면 또는 뒷면 중 정확히 하나를 선택해야 한다. 선택된 수가 왼쪽부터 오른쪽으로 갈수록 점점 커지도록 각 카드의 면을 선택하라.

각 카드 아래에 앞면 값과 뒷면 값이 표시되어 있다. 사용할 값을 클릭하면 해당 카드에 그 값이 선택된다. 다른 값을 클릭하면 선택이 변경된다. 인접한 두 카드의 선택된 값이 조건을 만족하지 않는 곳은 붉은색으로 표시된다.

“다시 하기” 버튼을 클릭하면 모든 카드를 처음 상태로 되돌릴 수 있다.

채점 기준

선택된 9개의 수가 왼쪽부터 오른쪽으로 갈수록 점점 커지면 전체 점수의 100%를 얻는다.

| | | | | | | | | |
|------|-------|------|-------|-------|-------|-------|-------|-------|
| #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 |
| ? | ? | ? | ? | ? | ? | ? | ? | ? |
| 11 8 | 14 10 | 12 9 | 18 21 | 23 19 | 22 26 | 28 25 | 24 27 | 29 27 |

각 카드에서 사용할 값을 선택하세요

다시 하기

14. 조 나누기 (9점)

1부터 16까지의 자연수가 있다. 이 수들을 4개의 조(가, 나, 다, 라)에 4개씩 나누어 넣으려고 한다.

각 조의 **점수**는 그 조에 속한 수 중 가장 작은 수와 가장 큰 수의 합이다. 예를 들어, 한 조에 {2, 5, 11, 14}가 있다면 그 조의 점수는 $2 + 14 = 16$ 이다.

전체 점수는 4개 조의 점수를 모두 더한 값이다.

전체 점수를 **최소화**하는 배치를 찾아라.

배치할 수를 클릭하여 선택한 뒤, 원하는 조의 빈 칸을 클릭하면 해당 위치에 배치된다. 배치된 수를 클릭하면 다시 되돌릴 수 있다. 각 조의 점수는 조 오른쪽에 표시되며, 하단에는 전체 점수가 표시된다.

“다시 하기” 버튼을 누르면 배치를 초기화할 수 있다.

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

| | | | | | | |
|---|----------------------|----------------------|----------------------|----------------------|---|---|
| 가 | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | = | ? |
| 나 | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | = | ? |
| 다 | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | = | ? |
| 라 | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | = | ? |

전체 점수: ?

15. 철도 (9점)

11개의 역과 18개의 선로로 이루어진 철도망이 주어진다. 각 역은 원으로 표시되어 있다. 당산역은 파란색, 잠실역은 주황색, 나머지 역은 흰색이다.

역과 역 사이를 잇는 양방향으로 통행 가능한 선로가 선으로 표시되어 있으며, 각 선로 옆에 철거 비용이 적혀 있다.

선로를 통해 당산역(파란색)에서 잠실역(주황색)으로 가는 모든 경로를 없애려고 한다. 여러분은 선로를 선택하여 철거할 수 있으며, 선택한 선로를 모두 철거했을 때 당산역에서 잠실역으로 갈 수 있는 경로가 없어야 한다.

선택된 선로는 빨간색 굵은 선으로 표시되며, 화면 하단에 현재 선택한 선로들의 총 비용이 표시된다.

조작 방법

- **선로 클릭:** 선로를 클릭하면 선택하거나 해제할 수 있다.
- **확인:** 선택한 선로들이 모든 경로를 차단하는지 검증한다. 차단하지 못하는 경우 남아 있는 경로가 노란색으로 표시된다.
- **초기화:** 모든 선택을 해제한다.

목표

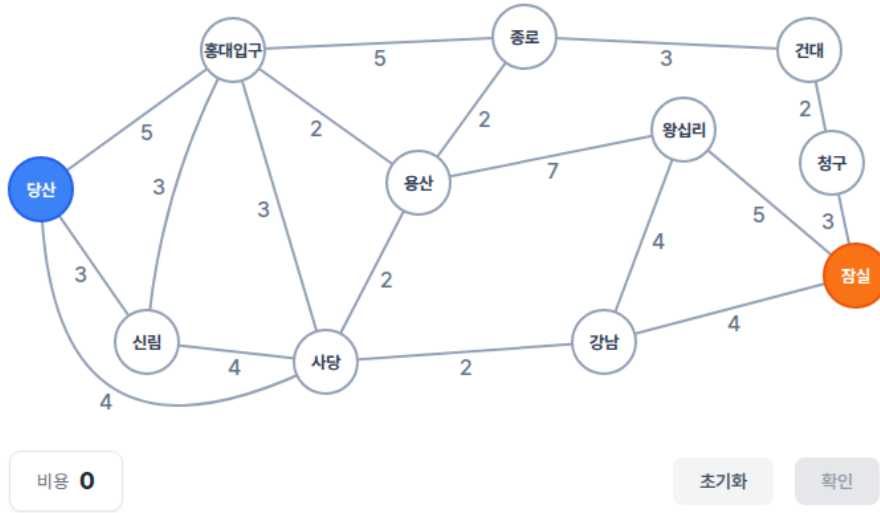
철거 비용의 합을 최소화하면서 당산역에서 잠실역으로 가는 모든 경로를 차단하는 것이다.

제약 조건

- 하나 이상의 선로를 선택해야 제출할 수 있다.

부분점수

1. (20%) 선택한 선로들이 당산역에서 잠실역으로 가는 모든 경로를 차단하면 전체 점수의 20%를 받는다.
2. (80%) 모든 경로를 차단하면서 철거 비용의 합이 최소이면 전체 점수의 나머지 80%도 받는다.



16. 직사각형 만들기 (14점)

12개의 막대기가 있고, 각 막대기가 아래에 있는 수평선에 놓여 있다. 각 막대기의 길이는 막대기에 적혀 있다.

이 막대기들을 두 묶음으로 나누어 직사각형을 만들려고 한다. 왼쪽 묶음에 속한 막대기들의 길이의 합이 직사각형의 세로가 되고, 오른쪽 묶음에 속한 막대기들의 길이의 합이 가로가 된다.

막대기들을 적절히 나누어, 만들어지는 직사각형의 넓이를 최대화하라. 빈 묶음의 경우 길이의 합은 0으로 간주한다.

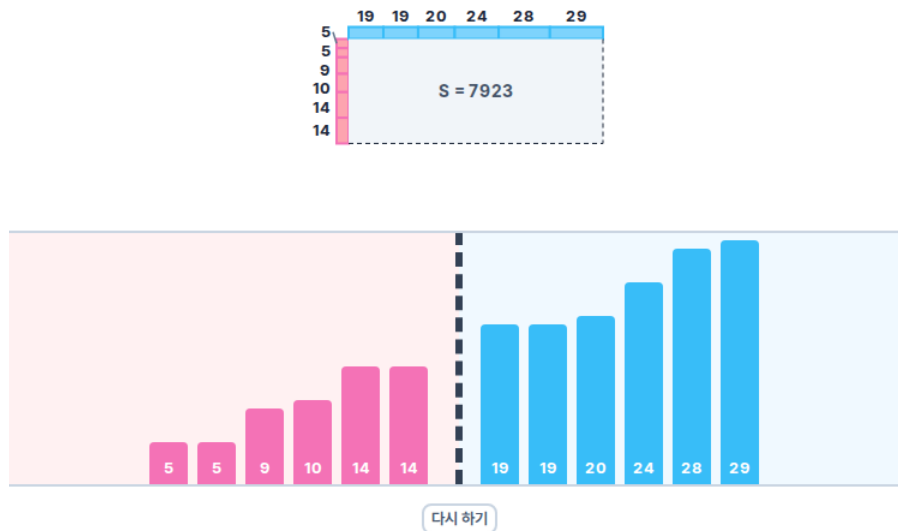
조작 방법

막대기들이 구분선을 기준으로 왼쪽 묶음(분홍색)과 오른쪽 묶음(하늘색)으로 나누어 표시된다. 각 막대기의 높이는 길이에 비례하며, 길이가 수로 표기되어 있다. 현재 묶음에 따라 만들어지는 직사각형의 모양이 상단에 표시된다.

막대기를 드래그하여 원하는 묶음으로 옮길 수 있다. 구분선을 넘겨 다른 묶음으로 이동시킬 수도 있다. “다시 하기” 버튼을 눌러 처음 상태로 되돌릴 수 있다.

채점 기준

직사각형의 넓이가 최댓값이면 전체 점수의 100%를 얻는다.



17. 스택 큐 정렬 (14점)

1부터 16까지의 정수로 이루어진 순열이 주어진다. 이 순열을 아래와 같이 일렬로 연결된 9개의 컨테이너를 이용하여 오름차순으로 정렬하려고 한다.

Input → Stack 1 → Queue 1 → Stack 2 → Queue 2 → Stack 3 → Queue 3 → Stack 4 → Output

각 컨테이너는 큐(Queue) 또는 스택(Stack) 중 하나의 방식으로 동작한다.

- 큐 (Input, Queue 1, Queue 2, Queue 3, Output): 맨 앞의 원소만 꺼낼 수 있다.
- 스택 (Stack 1, Stack 2, Stack 3, Stack 4): 맨 뒤의 원소만 꺼낼 수 있다.

꺼낸 원소는 바로 다음 컨테이너의 맨 뒤에 추가된다. Output에서는 원소를 꺼낼 수 없다.

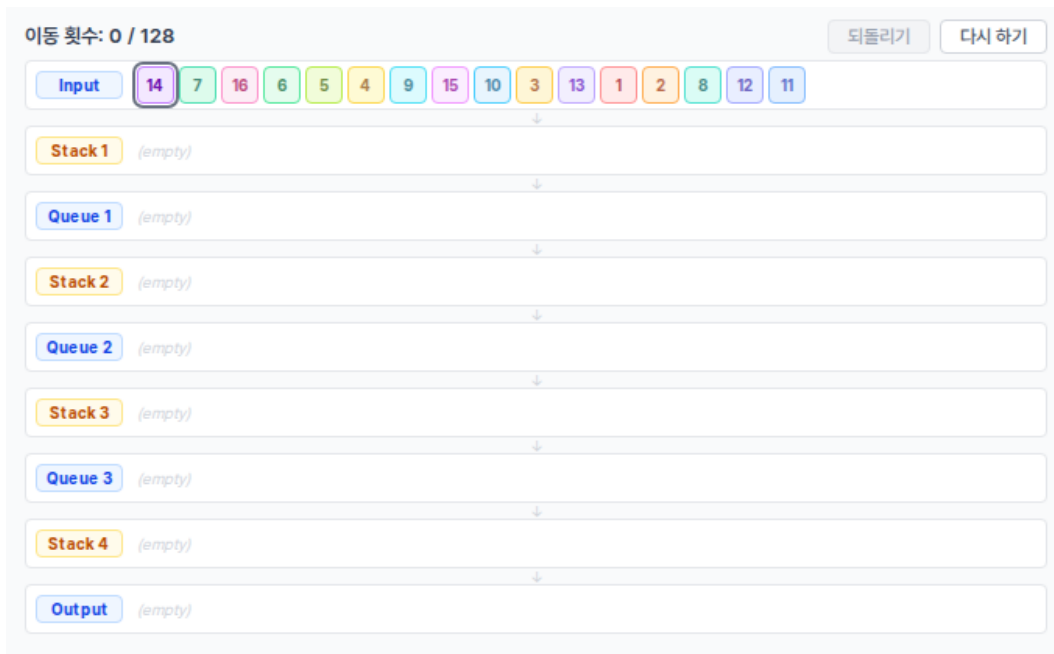
처음에 Input에는 [14, 7, 16, 6, 5, 4, 9, 15, 10, 3, 13, 1, 2, 8, 12, 11]이 들어 있고, 나머지 컨테이너는 비어 있다.

128회 이내의 이동으로 Output에 1, 2, 3, ..., 16이 순서대로 들어 있도록 하라.

꺼낼 수 있는 원소를 클릭하면 다음 컨테이너로 이동한다. 큐 컨테이너는 파란색, 스택 컨테이너는 주황색 라벨로 구분되어 있으며, 꺼낼 수 있는 원소는 테두리가 강조되어 표시된다. “다시 하기” 버튼을 누르면 처음 상태로 되돌릴 수 있다. “되돌리기” 버튼을 누르면 가장 최근 행동을 취소할 수 있다.

채점 기준

Output에 1부터 16까지 오름차순으로 정렬되어 있고, 이동 횟수가 128회 이하이면 전체 점수의 100%를 받는다.



18. 뽑기 (14점)

1부터 9까지의 수가 각각 두 번씩, 총 18개의 수가 일렬로 나열되어 있다.

여러분은 수열에서 인접한 두 수를 선택하여 제거하는 시행을 총 9번 수행해야 한다. 두 수를 제거하면 나머지 수들이 붙어 새로운 수열이 된다. 각 시행마다 제거한 두 수의 차이가 점수로 추가된다.

예를 들어, 1 2 3 8이 나열되어 있다고 하자. 1과 2를 제거하고, 3과 8을 제거하면 총점은 $1 + 5 = 6$ 이 된다. 그러나 2와 3을 먼저 제거하면 원래 인접하지 않았던 1과 8이 붙어 시행이 가능해지며, 1과 8까지 제거할 경우 총점은 $1 + 7 = 8$ 이 된다. 따라서 어떤 쌍을 먼저 제거하느냐에 따라 총점이 달라진다.

수를 클릭하여 선택하고, 그와 인접한 수를 클릭하면 두 수가 제거된다. 선택된 수는 파란색으로 강조 표시된다. ‘실행 취소’ 버튼을 통해 가장 마지막에 수행한 제거를 되돌릴 수 있다.

여러분의 목표는 9번의 시행을 모두 완료하여 총점을 최대화하는 것이다.

“다시 하기” 버튼을 누르면 처음 상태로 돌아간다.

채점 기준

9번의 시행을 모두 완료하고 총점이 이론적으로 가능한 최댓값이면 전체 점수의 100%를 받는다.

| | | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|------------|---|---|---|---|---|---|---|---|
| 1 | 3 | 2 | 4 | 1 | 5 | 3 | 2 | 4 | 5 | 6 | 8 | 7 | 9 | 6 | 8 | 7 | 9 |
| 총점: 0 | | | | | | | | | 남은 횟수: 9/9 | | | | | | | | |
| 제거 기록: (없음) | | | | | | | | | | | | | | | | | |
| 실행 취소 | | | | | | | | | 다시 하기 | | | | | | | | |

19. 양팔 저울 (16점)

양팔 저울과 10개의 추가 있다. 추의 무게는 각각 7, 10, 11, 12, 14, 16, 18, 20, 21, 22이다.

추를 하나씩 골라 왼쪽 또는 오른쪽 접시에 올리려고 한다. 아래와 같이 <와 >로만 이루어진 길이 10의 문자열이 주어진다.

<><><><>><

i 번째로 추를 올린 직후, 저울의 기울기가 문자열의 i 번째 문자와 일치해야 한다.

- > : 왼쪽 접시가 더 무거움 (왼쪽으로 기울어짐)
- < : 오른쪽 접시가 더 무거움 (오른쪽으로 기울어짐)

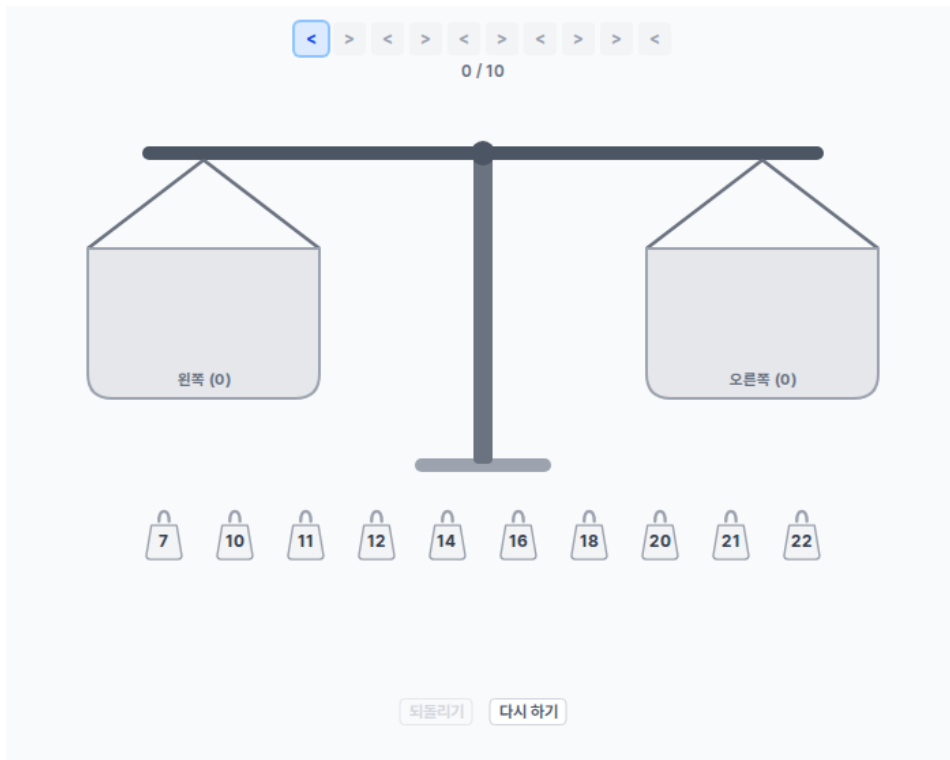
예를 들어, 첫 번째로 무게 7인 추를 오른쪽 접시에 올리면 오른쪽이 더 무거우므로 기울기는 <이다. 문자열의 첫 번째 문자가 <이므로 조건에 맞는다. 이어서 두 번째로 무게 22인 추를 왼쪽 접시에 올리면 왼쪽 총무게는 22, 오른쪽 총무게는 7이므로 기울기는 >이다. 문자열의 두 번째 문자가 >이므로 역시 조건에 맞는다.

이와 같이 10개의 추를 모두 올려서 매 단계마다 조건을 만족하도록 하라.

아직 올리지 않은 추 중 하나를 클릭하여 선택한 뒤, 왼쪽 또는 오른쪽 접시를 클릭하여 배치할 수 있다. “되돌리기” 버튼을 누르면 마지막 배치를 되돌릴 수 있고, “다시 하기” 버튼을 누르면 처음 상태로 되돌릴 수 있다.

채점 기준

10개의 추를 모두 올렸고, 매 단계마다 저울의 기울기가 주어진 문자열과 일치하면 전체 점수의 100%를 받는다.



20. 경로 세기 (17점)

시작 정점 S(파란색)와 끝 정점 E(주황색), 그리고 S에서 E로 향하는 간선 하나로 이루어진 방향 그래프가 주어진다. 현재 S에서 E로 가는 경로는 1개이다.

여러분은 새로운 정점을 추가하거나 정점 사이에 간선을 그어서 그래프를 확장할 수 있다. 단, 그래프에 사이클이 생기면 안 된다.

화면 상단에 S에서 E까지의 현재 경로 수가 큰 숫자로 표시된다. 목표를 달성하면 초록색으로 바뀐다. 정점은 원으로 표시된다. 시작 정점 S는 파란색, 끝 정점 E는 주황색, 추가된 정점은 흰색이다. 간선은 화살표로 표시되며, 방향을 나타낸다. S→E 간선은 삭제할 수 없다.

조작 방법

- 간선 생성: 정점을 클릭하여 선택한 뒤, 다른 정점을 클릭하면 첫 번째 정점에서 두 번째 정점으로 향하는 간선이 만들어진다.
- 간선 삭제: 이미 존재하는 간선 위에서 같은 조작을 하거나, 간선을 직접 클릭하면 삭제된다. S→E 간선은 삭제할 수 없다.
- 정점 추가: 하단의 + 버튼을 클릭하면 새 정점이 추가된다. 정점은 최대 20개까지 추가할 수 있다.
- 정점 삭제: S와 E를 제외한 정점을 휴지통 아이콘으로 드래그하면 삭제된다.
- 정점 이동: 정점을 드래그하여 위치를 조정할 수 있다.

목표

S에서 E까지의 서로 다른 경로 수가 정확히 2026이 되도록 그래프를 구성하는 것이다.

제약 조건

그래프에는 사이클이 있으면 안 된다. 간선을 생성할 때 사이클이 생기는 경우, 생성할 수 없다.

채점 기준

S에서 E까지의 경로 수가 정확히 2026이면 전체 점수를 받는다.

1

