10주차

정렬 알고리즘

인하공업전문대학 컴퓨터정보과

이수정 교수

지난시간

- 프로그램 논리
  1. 순차 논리
  2. 선택 논리
  3. 반복 논리
     - while, do-while, for문
     - 다중반복문

1. 반복 논리

- **반복의 의미**
  - 동일한 처리 과정이 여러 번 수행되는 것
- **반복 논리**
  - 컴퓨터 프로그램으로 반복을 해결하기 위한 논리
  - 어떤 조건에 따라서 몇몇 특정의 문장들을 반복수행하기 위한 논리
- **반복 논리를 위한 제어문**
  - while문
  - do-while문
  - for문

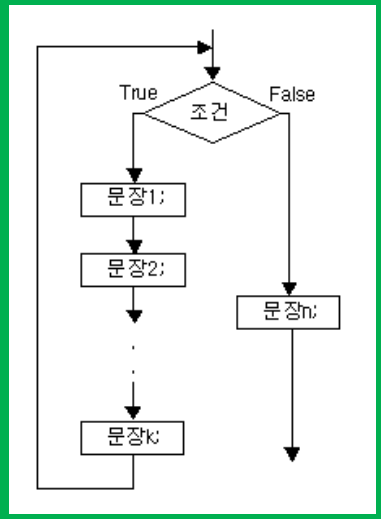
2.1 while문

- 주어진 조건이 만족되는 동안(즉, 주어진 조건이 참[true]인 동안)만 while문 내의 문장들을 반복해서 수행
- while문의 조건에 대한 검사가 while문 내의 문장들을 수행하기 전에 행해짐
- 조건 검사가 어느 위치에서 수행되는지는 프로그램의 논리에 있어 매우 중요

2.1 while문

- **while문의 형식**

형식1	순서도

while(조건) {
문장1;
문장2;
...
문장k;
}
문장n;
	

- while 블록 안의 문장들(문장1, 문장2, . . 문장k)을 조건이 거짓이 될 때까지 수행
- while 블록과 while문의 조건을 하나로 묶어 while 루프(loop)라고 부름
- 처음에 while문의 조건을 검사했을 때, 거짓이 되면 while 블록 내의 어떤 문장도 수행하지 않음
- 루프 내에는 while의 조건을 거짓으로 만들기 위한 문장이 들어가 있음
- 만약 while 루프 내에서 while의 조건을 거짓으로 만들지 못하면 while문은 무한히 반복하게 됨

확인문제(1)

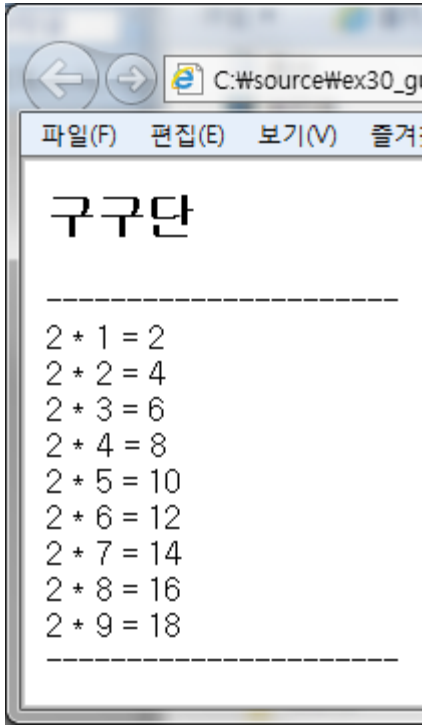
- **다음과 같이 구구단 중에서 2단을 출력하는 순서도와 프로그램을 while문을 이용하여 작성하여라.**

(문제해결논리)

구구단의 2단에서 2는 같은 숫자로 고정되어 있다. 고정된 숫자 2에 1부터 9까지 count를 하나씩 증가시키면서 곱해주면 된다.

아홉 번 과정이 반복된다. 따라서 고정된 값 2를 저장하는 변수 dan과 증가 값 count 변수를 선언하고 초기 값을 각각 2와 1로 부여한다.

그런 다음 while문을 사용하여 반복수행하도록 한다. while문의 조건은 증가 값 count가 9가 되었는지를 검사하면 된다. 프로그램의 종료는 count가 9일 때도 구구단을 계산해야 하므로 count가 10이 되면 종료한다.



확인문제(2)

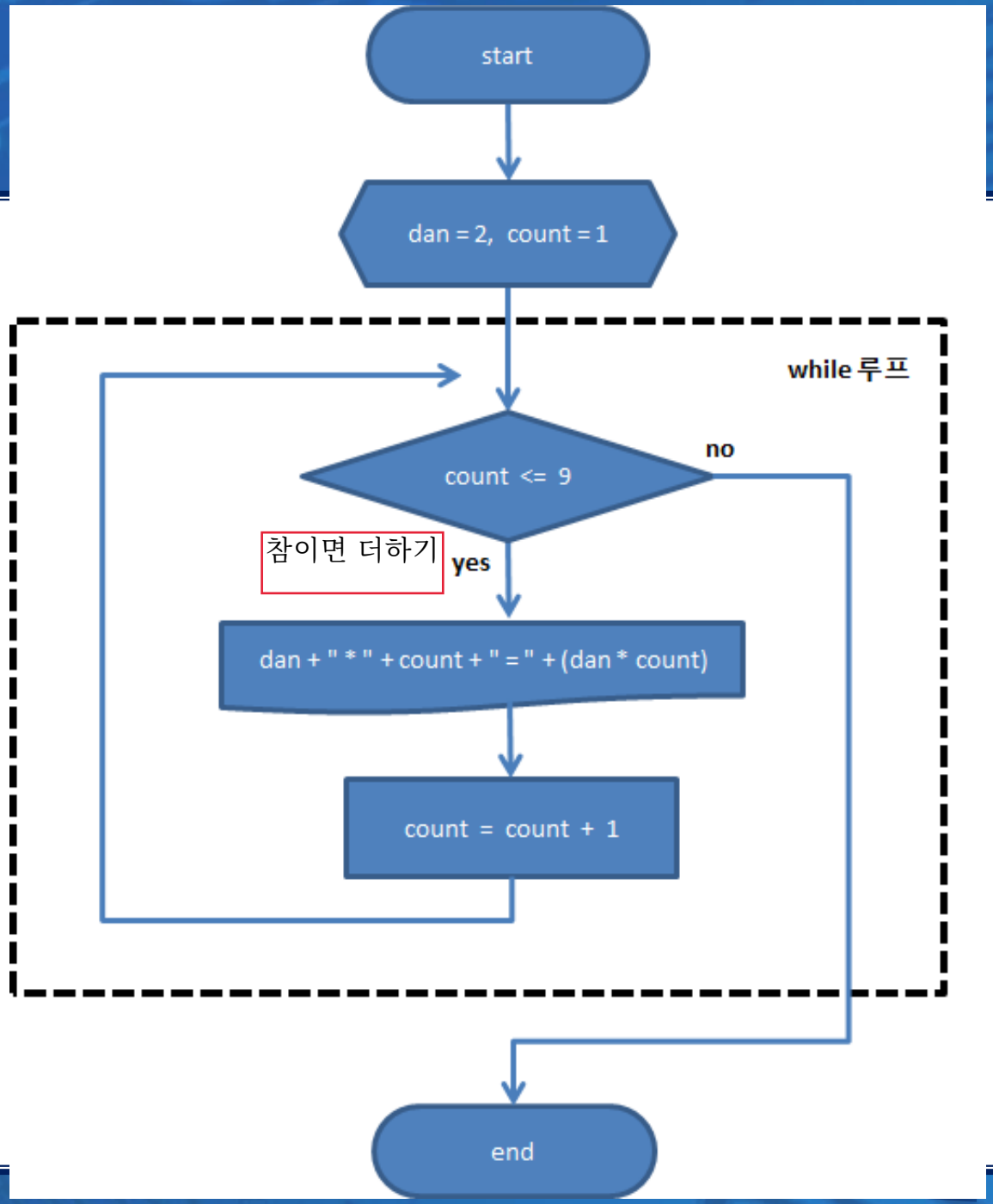
ex30_gugudan - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
<html>
<head> <title> 구구단 </title> </head>
<body>
   <script type = "text/javascript">
            document.write("<h2>구구단</h2>");
            document.write("---------------------<br>");

            var  dan = 2; count = 1;

            while(count <= 9) {
               document.write(dan + " * " + count + " = " + (dan * count));
               document.write("<br>");
               count = count + 1;
            }
            document.write("---------------------<br>");
     </script>
  </body>
</html>
```



이번 시간

- 정렬 알고리즘 (Sorting Algorithm)

차례

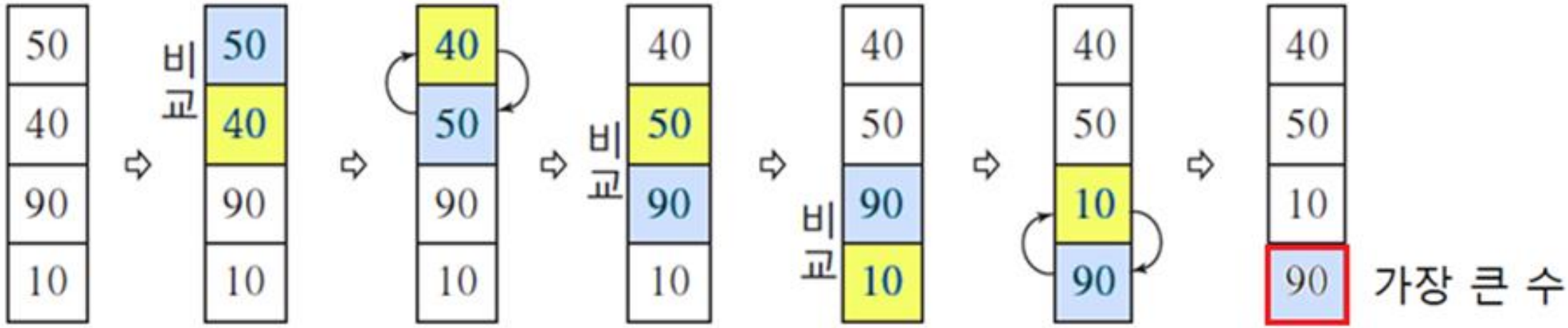
- 버블 정렬
- 칵테일 정렬
- 삽입 정렬
- 팬케이크 정렬

개요

- 정렬 : 데이터를 특정한 조건에 따라 일정한 순서가 되도록 다시 배열(arrange)하는 일
- 배열(array) : 정렬할 숫자들이 일렬로 저장된 곳

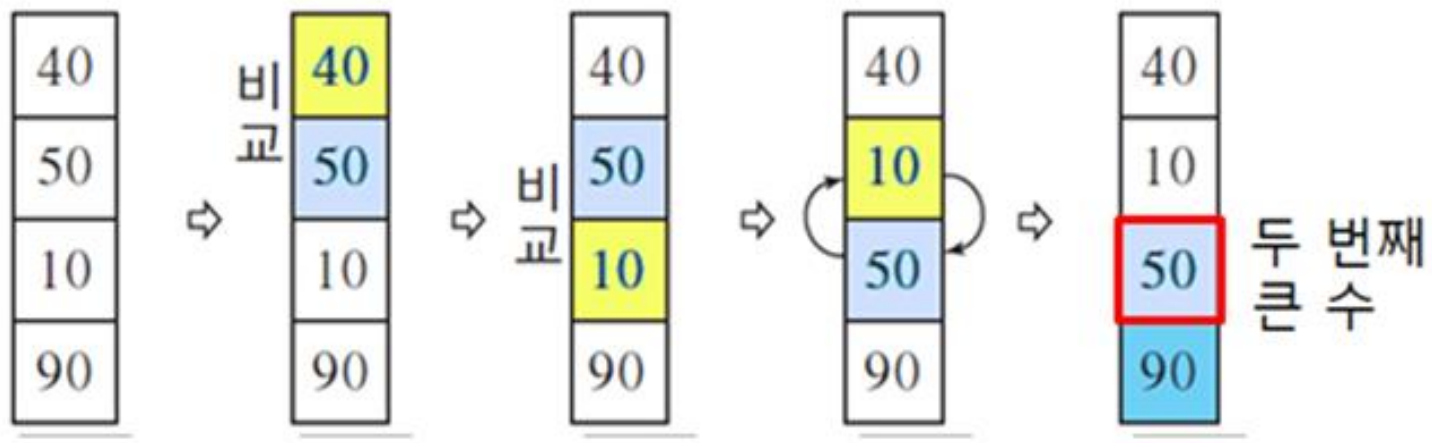
버블 정렬

- 버블 정렬(Bubble sort) : 이웃하는 숫자를 비교하여 작은 수를 앞쪽으로 이동시키는 과정을 반복하여 정렬한다.

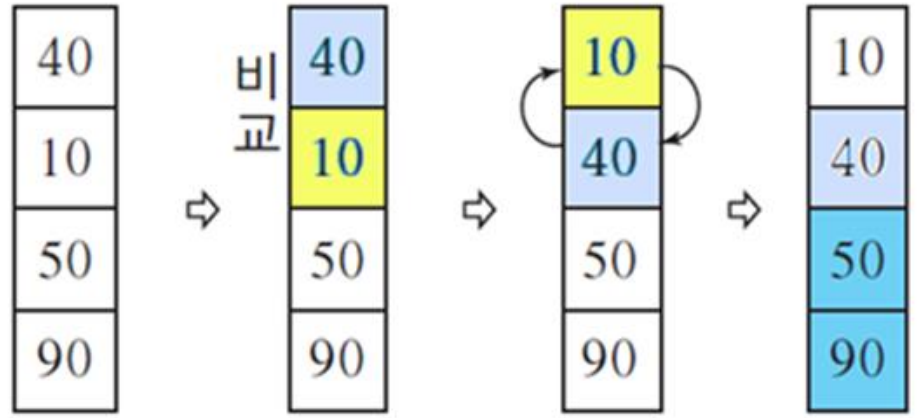


버블 정렬

- 두 번째 패스



- 세 번째 패스



버블 정렬

- 오름차순 상하 정렬로 생각해보면, 작은 수가 마치 거품(버블, bubble)처럼 위로 올라가는 것을 연상케 한다.
- n개의 숫자가 있으면 최대 (n-1)번의 패스가 수행된다.

칵테일 정렬

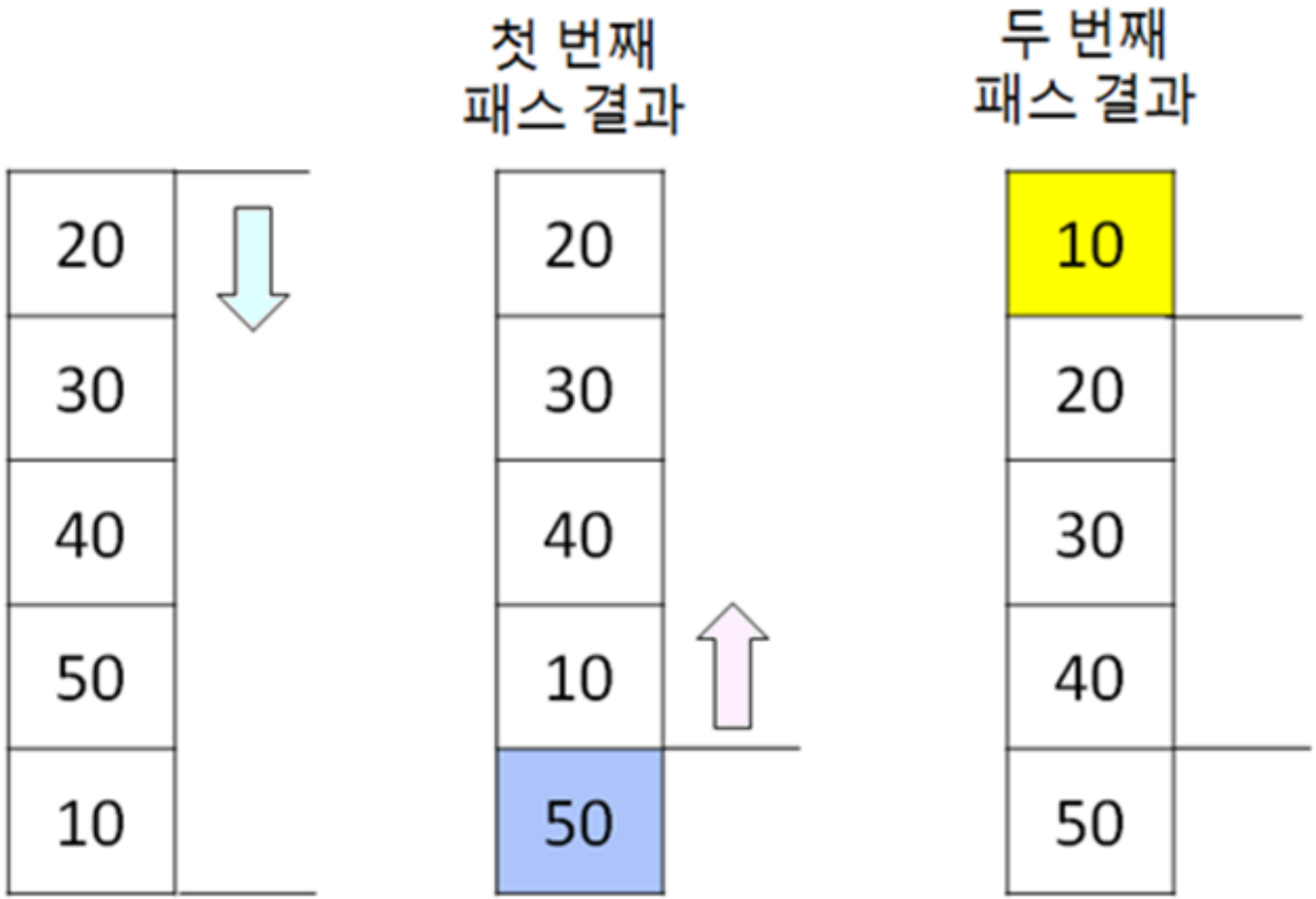
- 버블 정렬에서 가장 큰 수는 토끼같이 빠르게 아래쪽으로 내려가서 제자리를 잡지만, 작은 수들은 기껏해야 한 칸씩 거북이처럼 위로 올라간다.
- 칵테일 정렬은 작은 수들에게도 토끼가 될 기회를 주자는 것이다.

칵테일 정렬 - 알고리즘

- 칵테일 정렬은 홀수 번째 패스에서 버블 정렬처럼 위에서 아래 방향으로 이웃한 숫자들을 비교하며 위에 있는 숫자가 크면 서로 교환한다. 그리고 짝수 번째 패스에서는 아래에서 위 방향으로 이웃한 숫자를 비교하여 아래에 있는 숫자가 작으면 서로 교환한다.

- 마치 칵테일을 만들 때 위 아래로 흔드는 것과 같다고 해서 정렬 이름을 칵테일 정렬이라 부른다.

칵테일 정렬



칵테일 정렬

- 버블 정렬을 개선한 알고리즘이다.
- 첫번째 패스가 수행된 후에는 가장 큰 숫자가 배열 (리스트)의 마지막 원소에 자리 잡는다.
- 두번째 패스에는 가장 작은 숫자가 배열(리스트)의 첫 원소에 자리 잡는다.

칵테일 정렬

- [문제] 입력이 [8, 6, 2, 5, 3, 9, 1, 7]일 때 칵테일 정렬 알고리즘의 3번째 패스가 종료된 결과는?

① [6, 2, 5, 3, 8, 1, 7, 9]

② [1, 6, 2, 5, 3, 8, 7, 9]

③ [1, 3, 5, 2, 6, 7, 8, 9]

④ [1, 2, 5, 3, 6, 7, 8, 9]

삽입 정렬

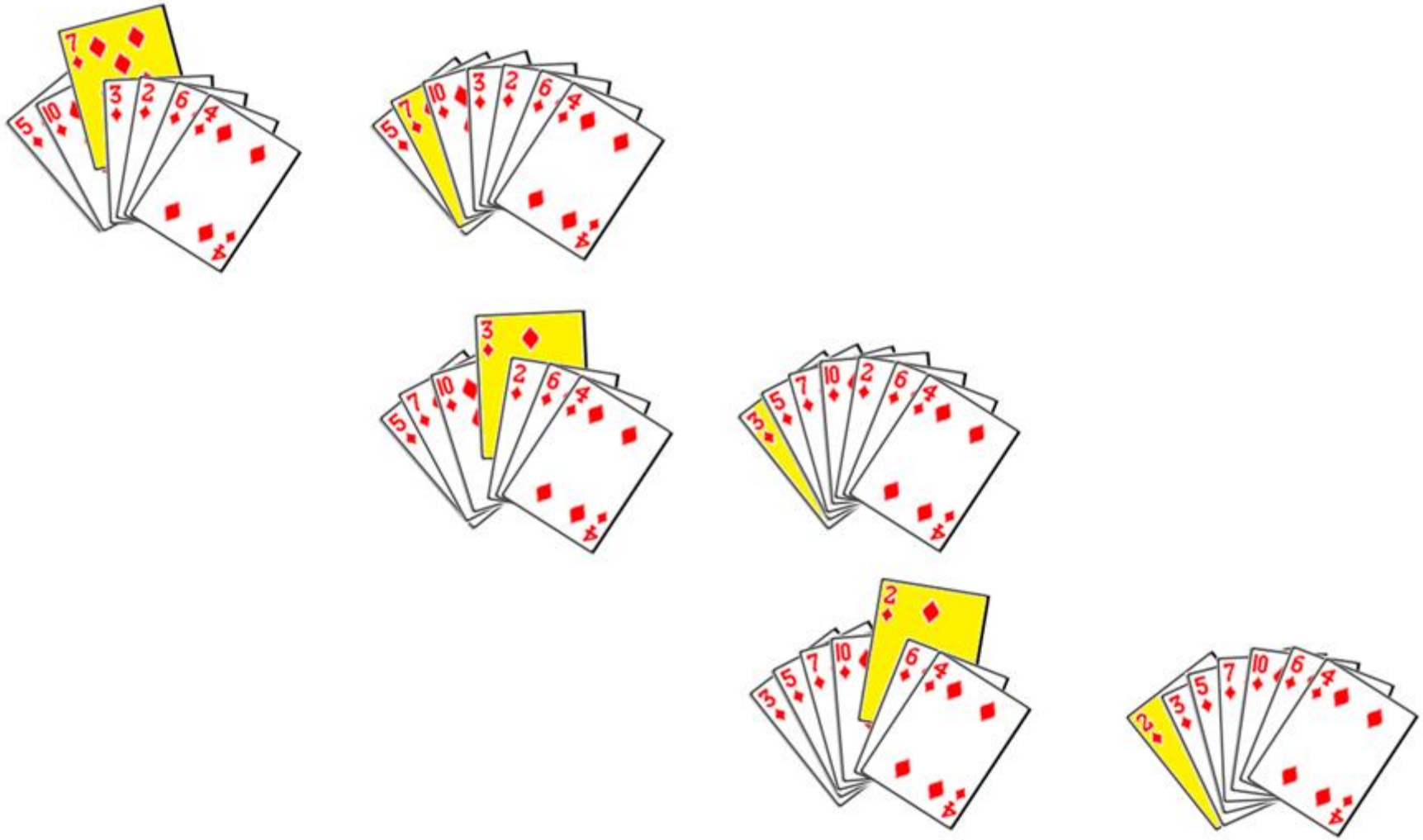
- 삽입 정렬(Insertion sort)
  - 나열된 숫자들의 앞부분은 정렬된 상태를 유지하면서 정렬이 안 된 뒷부분의 가장 왼쪽 숫자를 앞부분의 적절한 위치에 삽입하여 정렬되도록 하는 과정을 반복한다.

삽입 정렬


정렬된 부분
정렬 안 된 부분
40 60 70 50 10 20 30
50
40 60 70 10 20 30
비교
50
이동
40 60 70 10 20 30
50
40 60 70 10 20 30
비교
50
이동
40 60 70 10 20 30
50
40 60 70 10 20 30
비교
삽입
40 50 60 70 10 20 30
정렬된 부분
정렬 안 된 부분

삽입 정렬

삽입 정렬



삽입 정렬

삽입 정렬 – 알고리즘

1) 두번째 숫자를 현재 원소로 하여 시작한다.

2) 현재 원소를 왼쪽 방향의 숫자와 하나 씩 비교하여 현재 원소가 삽입될 곳을 찾아서 그 곳에 삽입한다.

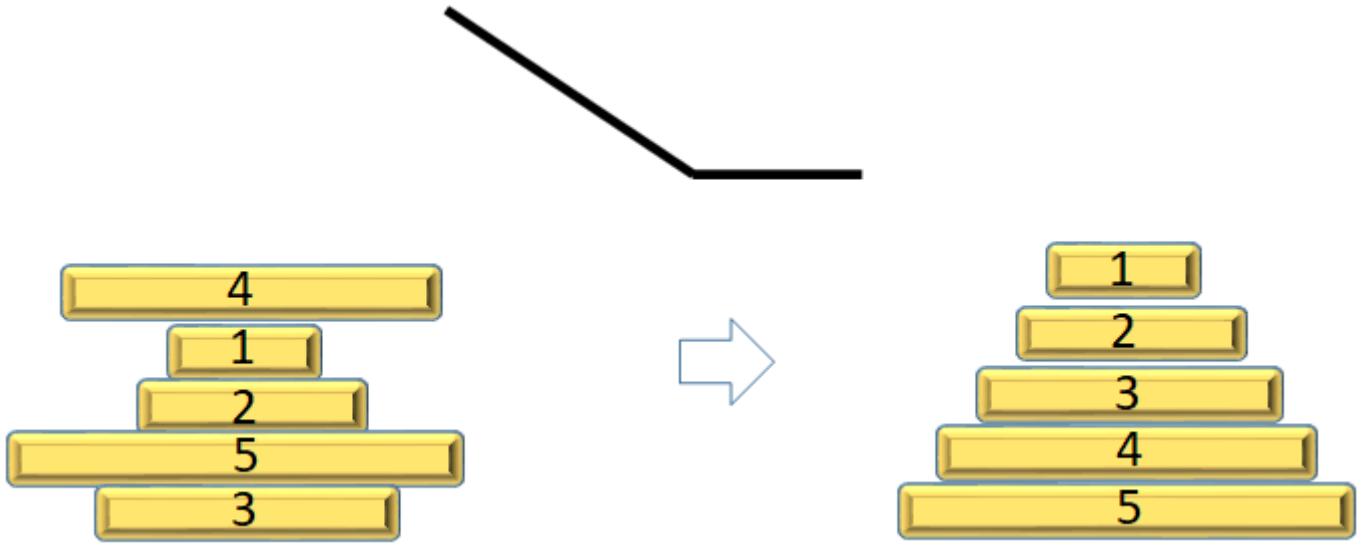
3) 이제 새 현재 원소는 직전 현재 원소가 있었던 곳의 바로 오른쪽 옆의 숫자이며, [2]를 수행한다.

팬케이크 정렬

- 팬케이크 정렬은 접시에 담긴 크기가 제각각인 팬케이크들을 뒤집개를 사용하여 크기 순서로 정렬하는 것이다.

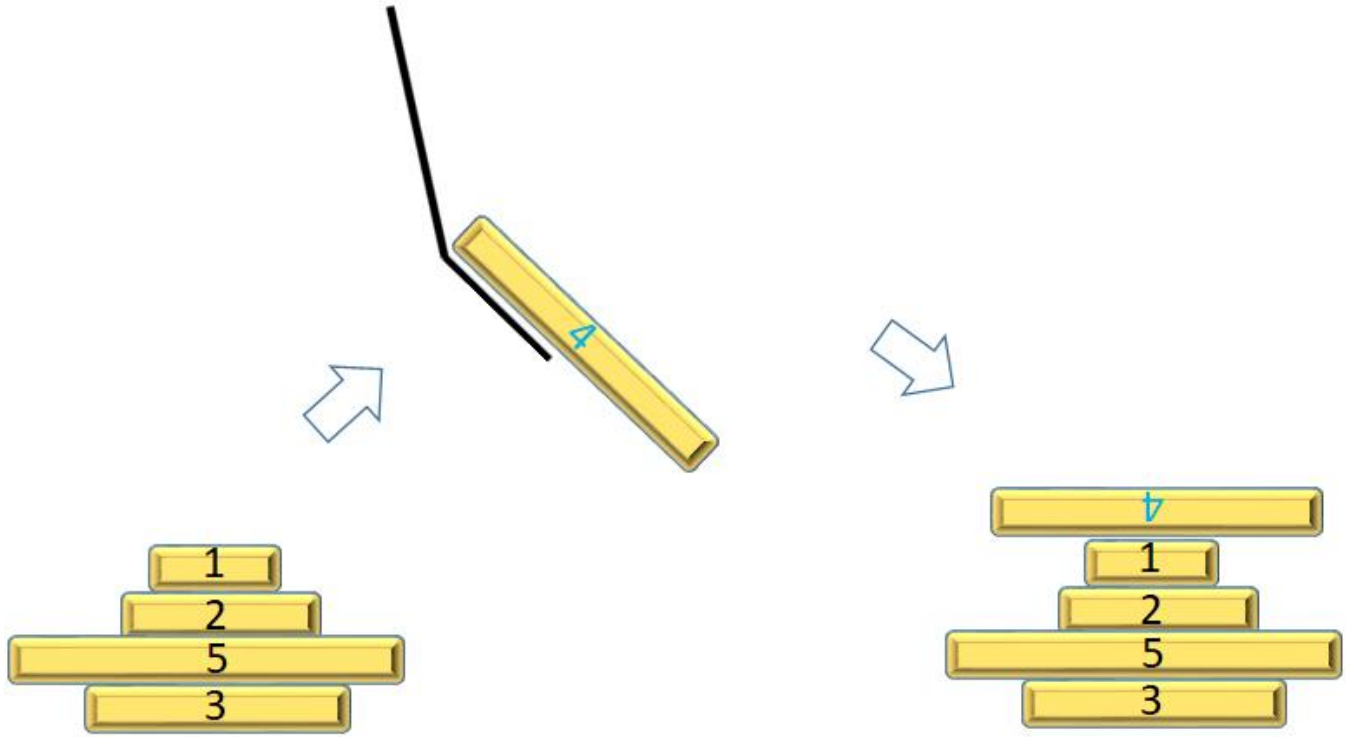
- 뒤집개는 팬케이크 사이에 끼울 수 있으며, 들어 올린 팬케이크 전체를 역순으로, 즉 뒤집어 놓을 수 있다.

팬케이크 정렬



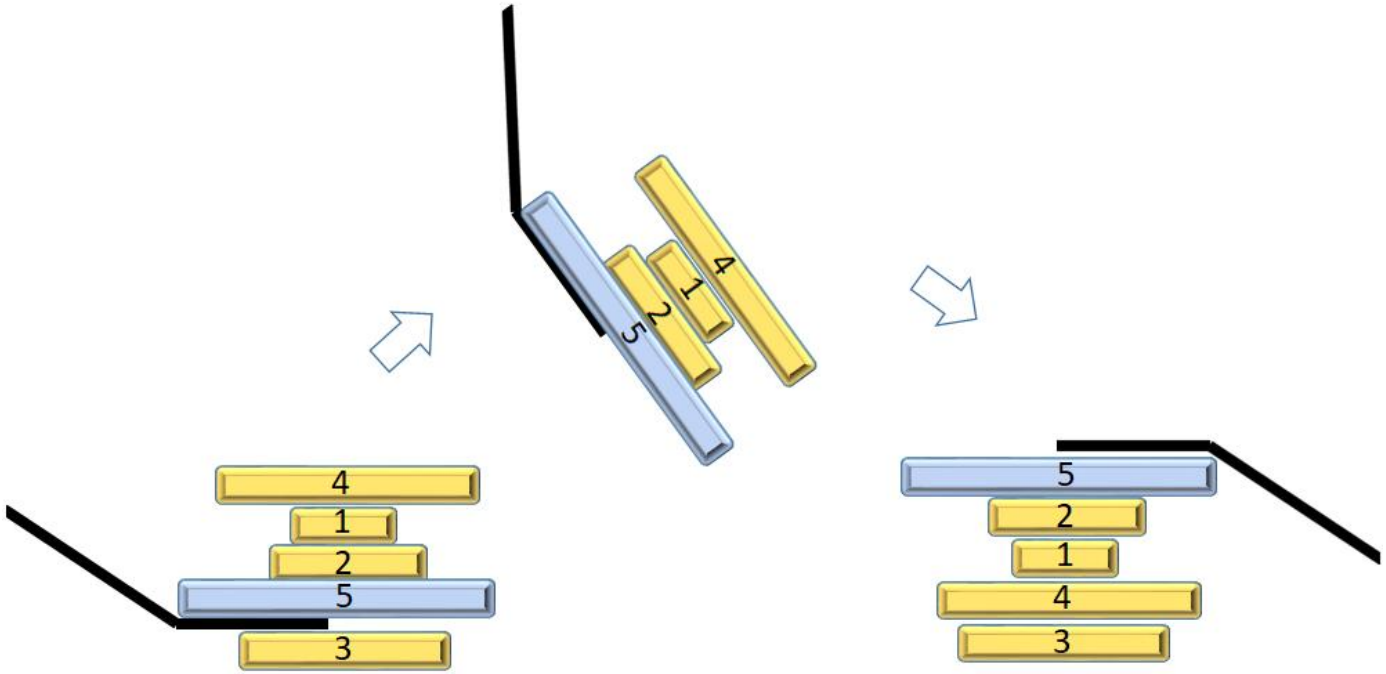
팬케이크 정렬

- 맨 위에 있는 것을 들어올려볼까



팬케이크 정렬

- 가장 큰 것을 들어올려볼까

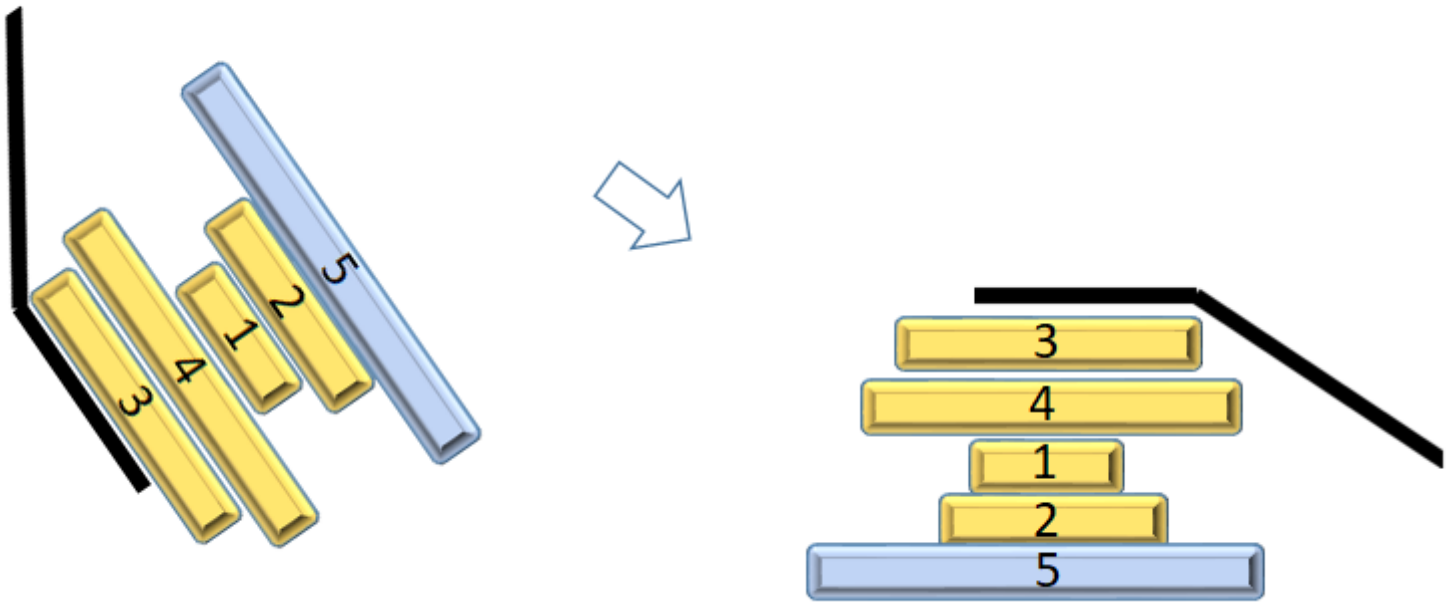


팬케이크 정렬 - 알고리즘

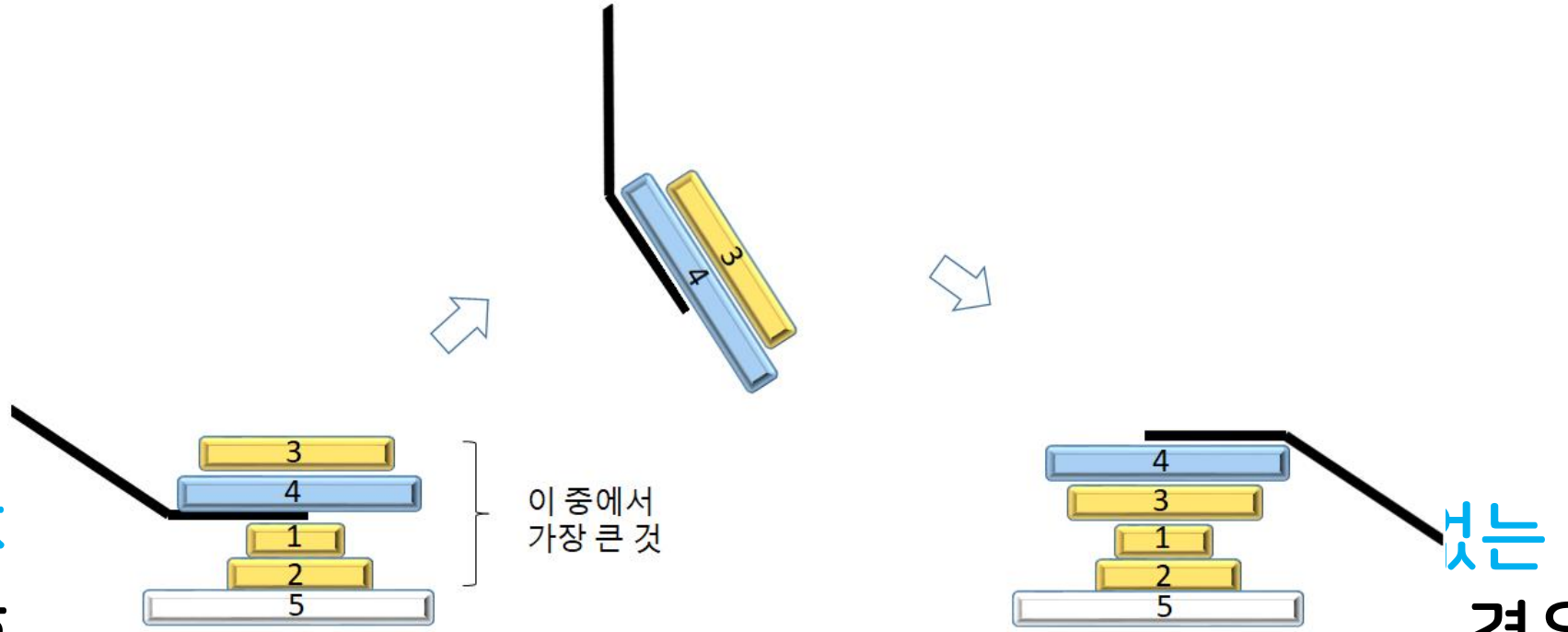
[1] 아직 정렬 안 된 팬케이크들 중에서 가장 큰 것을 찾아서 그 밑에 뒤집개를 끼워 넣고 들어 올려서, 180° 뒤집어 가장 큰 것이 맨 위에 오도록 만든다.

[2] 정렬이 아직 안 된 팬케이크들 중에서 맨 아래에 있는 팬케이크 바로 밑에 뒤집개를 끼워 넣고 들어올려서 180° 뒤집어 내려놓는다.

팬케이크 정렬



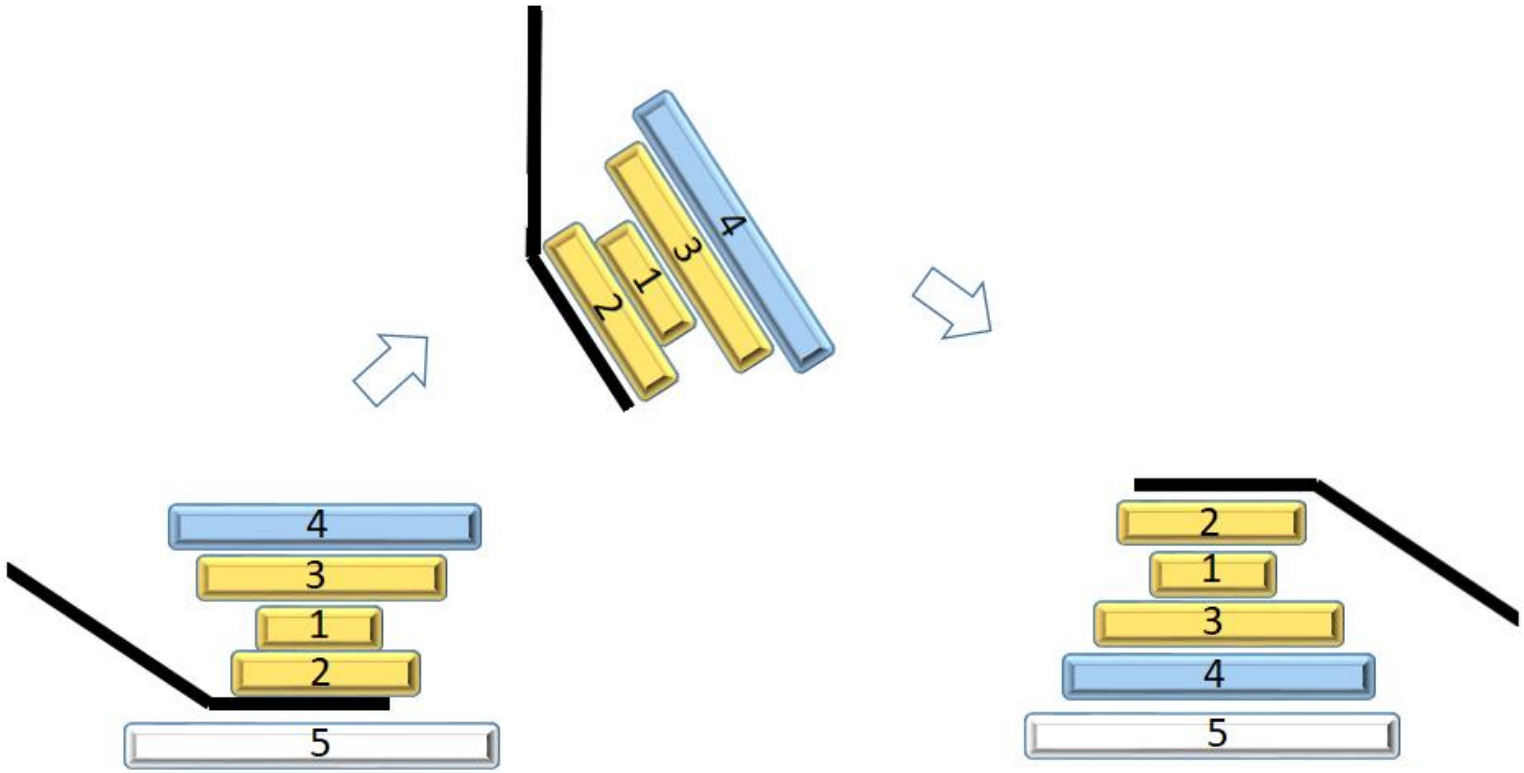
팬케이크 정렬



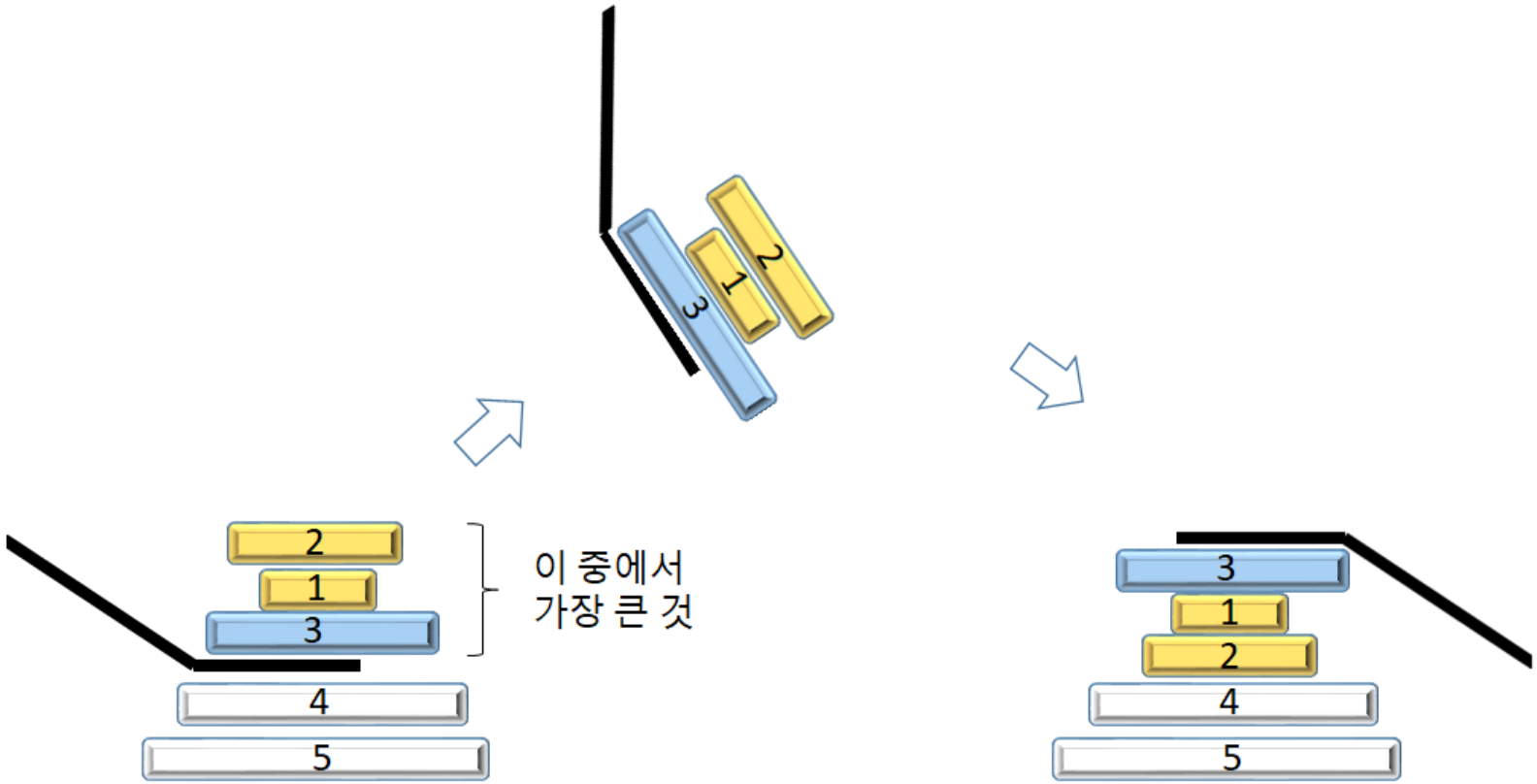
- 한번 제ㅈ
로 간주한 ... 있는 것으
다.

경우와 같

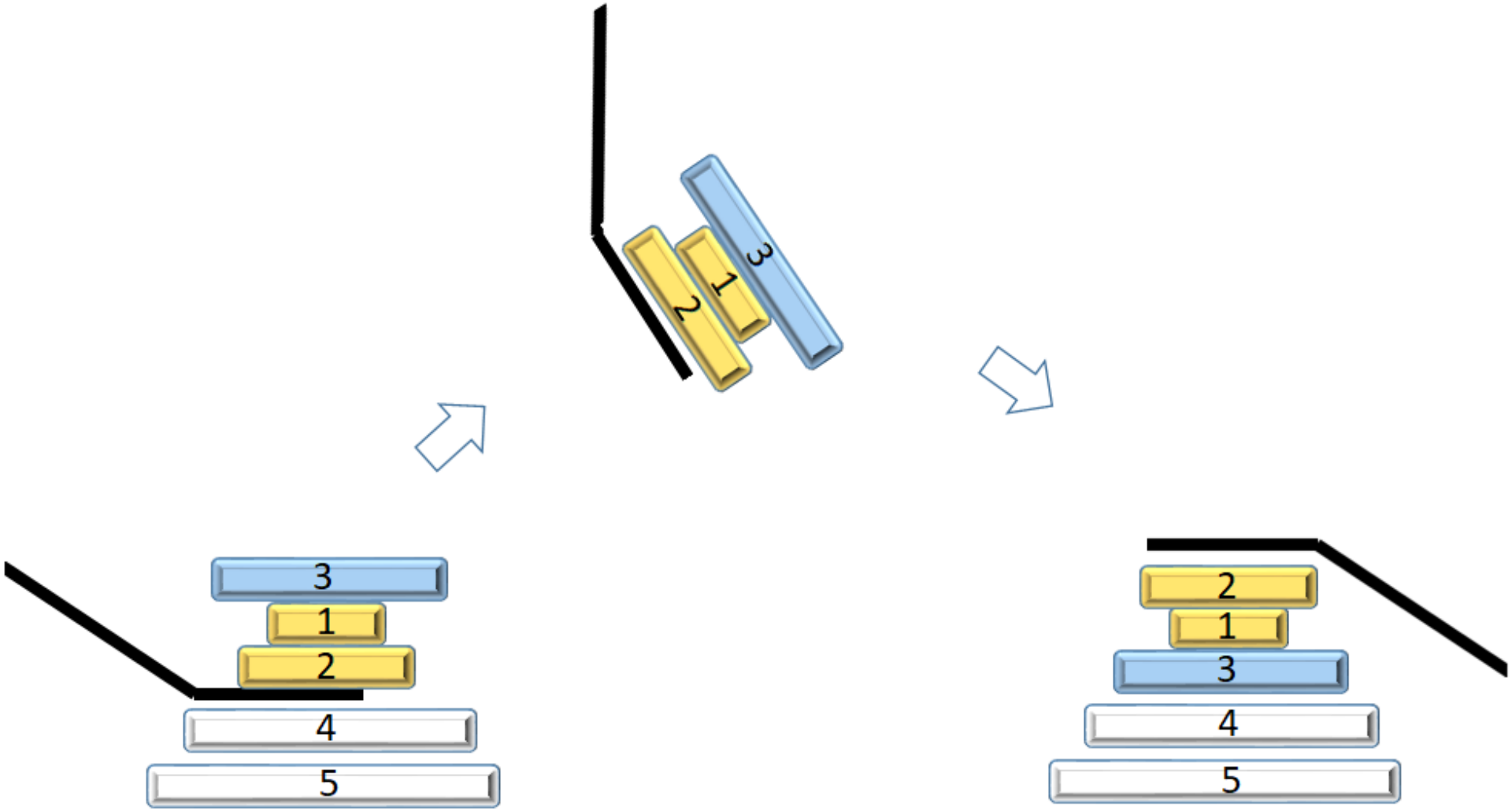
팬케이크 정렬



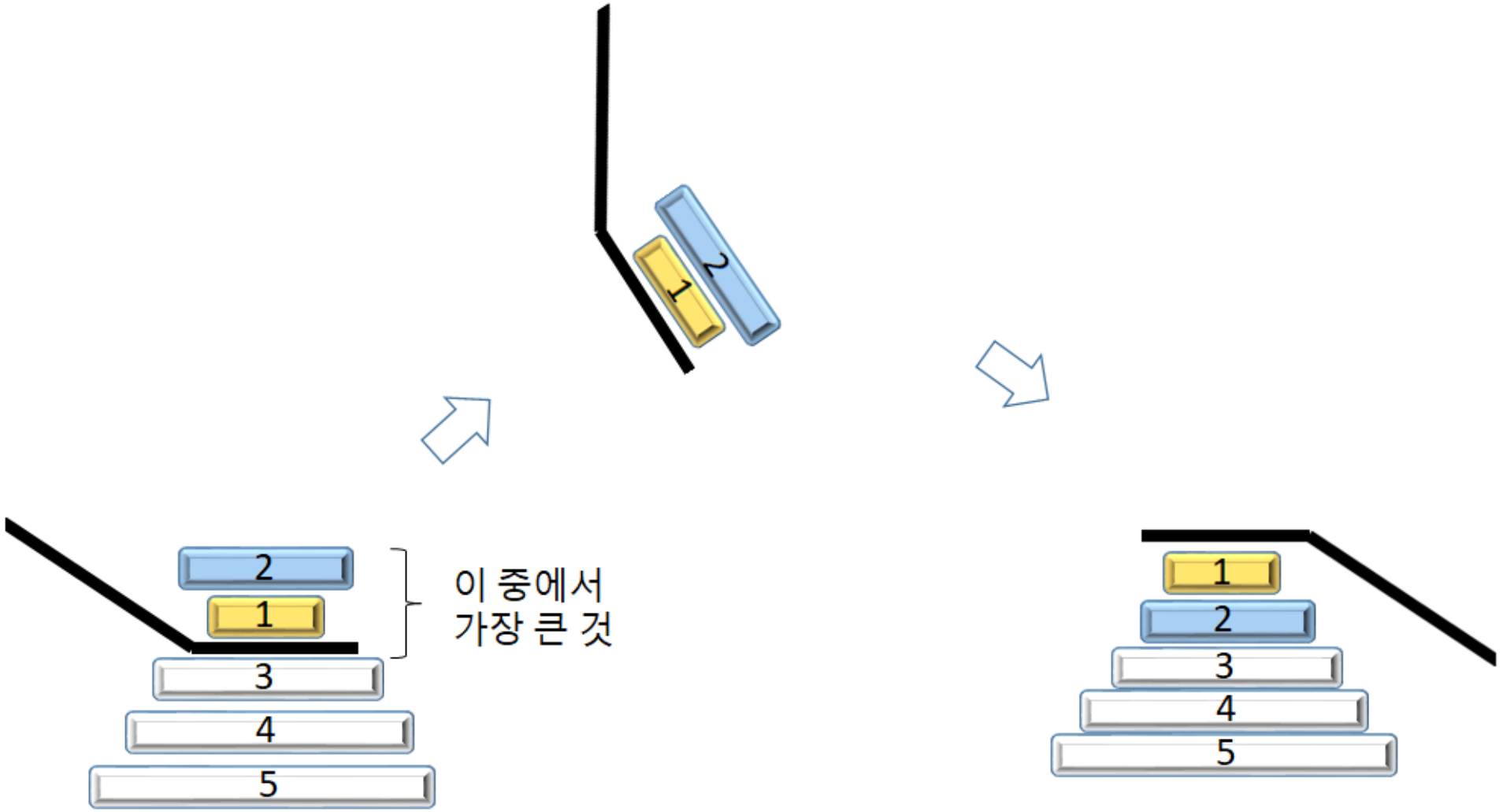
팬케이크 정렬



팬케이크 정렬



팬케이크 정렬



팬케이크 정렬

- [문제] 입력 [1, 5, 4, 3, 2]에 대해 팬케이크 정렬 알고리즘으로 정렬할 때 필요한 뒤집개의 사용 횟수는?

① 3회

② 4회

③ 5회

④ 6회

[정답] 4회

[5, 1, 4, 3, 2]

[2, 3, 4, 1, 5]

[4, 3, 2, 1, 5]

[1, 2, 3, 4, 5]

요약

- 지금까지 알려진 정렬 알고리즘은 수십 여개에 이른다.
- 이 중에서 실세계에 사용되는 정렬 알고리즘은 손가락에 꼽을 정도이다.
- 간단하고 재미있는 몇 개의 정렬 알고리즘만을 살

요약

- 버블 정렬: 이웃하는 것 바꾸기
- 칵테일 정렬: 양방향으로
- 삽입 정렬: 앞에다 삽입하기
- 팬케이크 정렬: 가장 큰 것 들어올려

다음 시간

11주차 : 다중 반복문