

알고리즘과게임콘텐츠

9장 자료탐색기법_순차탐색

학습 목표

1. 탐색 알고리즘의 개념을 이해한다.
2. 실생활에서 탐색알고리즘을 이용하여 문제해결을 할 수 있다.
3. 순차탐색을 파이선으로 구현할 수 있다.
4. 단순히 문제 풀기 알고리즘의 장단점을 이해할 수 있다.

탐색 (SEARCH) 의 필요성

- 순서대로 읽고 나를 찾아 보세요.
 - 나는 안경을 쓰지 않았습니다.
 - 나는 줄무늬 티셔츠를 입지 않았습니다.
 - 나는 모자를 쓰지 않았습니다.
 - 나는 양말을 신지 않았습니다.
 - 나는 치마를 입지 않았습니다.



탐색 (SEARCH) 의 필요성

- 우리가 수행해야 할 작업을 처리하거나 원하는 목적을 달성하기 위한 작업

- 도서관 책 중 원하는 책 찾기
- 영어 사전에서 단어 찾기
- 저렴한 가격에 좋은 물건 찾기
- 네비게이션의 경로 찾기

The screenshot displays the Sangmyung University Library website. At the top, there is a navigation bar with links for '로그인' (Login), 'ENGLISH', '블로그' (Blog), '천안 캠퍼스' (Cheonan Campus), '학술정보관' (Academic Information Center), and '상명대' (Sangmyung University). Below this is a main menu with '자료검색' (Data Search), '연구지원' (Research Support), '서비스' (Service), '커뮤니티' (Community), 'My Library', '학술정보관 안내' (Academic Information Center Guide), and 'Q&A'. The '자료검색' (Data Search) section is active, showing a sidebar with various search options like '소장자료검색' (Collection Search), 'DB통합검색' (DB Integrated Search), '전자저널AtoZ' (Electronic Journals AtoZ), '교내/교외 학위논문' (In-campus/Out-campus Theses), 'e-Book', '오디오북' (Audiobook), 'e-Learning', '모바일서비스' (Mobile Service), '상명사료' (Sangmyung University Archives), and '학술정보분석서비스' (Academic Information Analysis Service). The main search area is titled '상세검색' (Detailed Search) and includes a breadcrumb trail '홈 / 자료검색 / 상세'. The search form contains several fields: 'Q 캠퍼스 선택' (Campus Selection) with a dropdown set to '서울캠퍼스' (Seoul Campus); 'Q 검색항목' (Search Item) with a dropdown set to '서명' (Title); 'Q 검색항목' (Search Item) with a dropdown set to '저자명' (Author Name); 'Q 검색항목' (Search Item) with a dropdown set to '출판사' (Publisher); and '자료유형' (Resource Type) with a dropdown set to '전체' (All). There are also input fields for '검색어를 입력하세요.' (Enter search terms) and a dropdown for 'and'. At the bottom, there is a '발행년도 (시작)' (Publication Year (Start)) field with a range from 2000 to 2020. A 'Q 검색' (Search) button is located at the bottom right of the search form.

- 원하는 일을 보다 효율적으로 처리하기 위해 상황에 맞는 적절한 탐색 방법을 선택

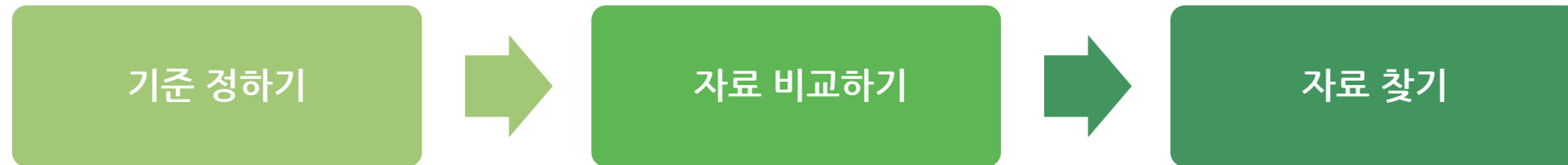
탐색의 개념

- 여러 개의 다양한 자료 중에서 기준에 적합한 자료를 찾는 것을 탐색이라 함
 - 필요한 자료나 정보를 찾는 일
 - 기준 값에 적합한 자료를 찾아내는 것



탐색의 개념

- 탐색의 과정

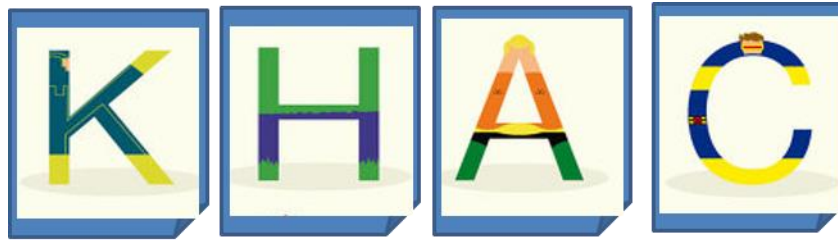


- 기준 정하기 : 무엇을 찾을 것인가 기준 설정하기
- 자료 비교하기 : 기준에 맞추어 자료를 비교하기
- 자료 찾기 : 기준과 일치하는 자료 찾기

탐색의 개념

■ 정렬과 탐색의 관계

찾는 값 : F



자료가 정리가 안 경우 : 끝까지 검색해야 F가 존재하지 않음을 알 수 있음

찾는 값 : F



자료가 정리가 된 경우 : H까지만 검색하면 F가 존재하지 않음을 알 수 있음

순차탐색(SEQUENTIAL SEARCH) 개념

- 모든 자료를 처음부터 끝까지 하나씩 비교하여 조건과 일치하는 자료를 찾는 알고리즘
- 까망머리 친구 찾기

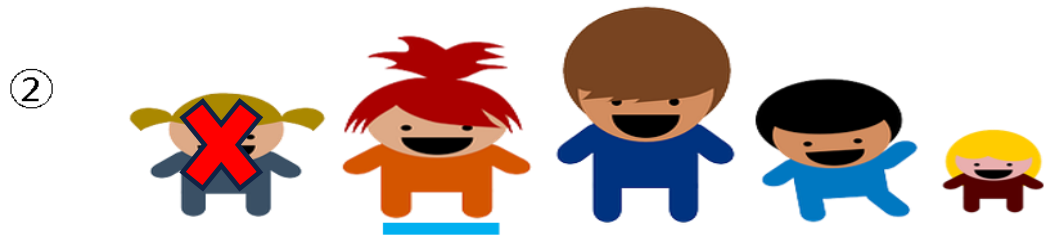
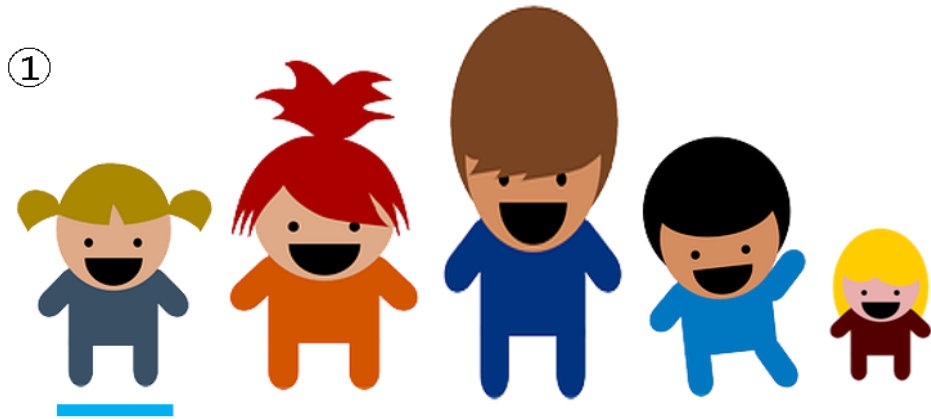


순차탐색(SEQUENTIAL SEARCH) 개념

- 까망머리 친구 찾기



순차탐색(SEQUENTIAL SEARCH) 개념

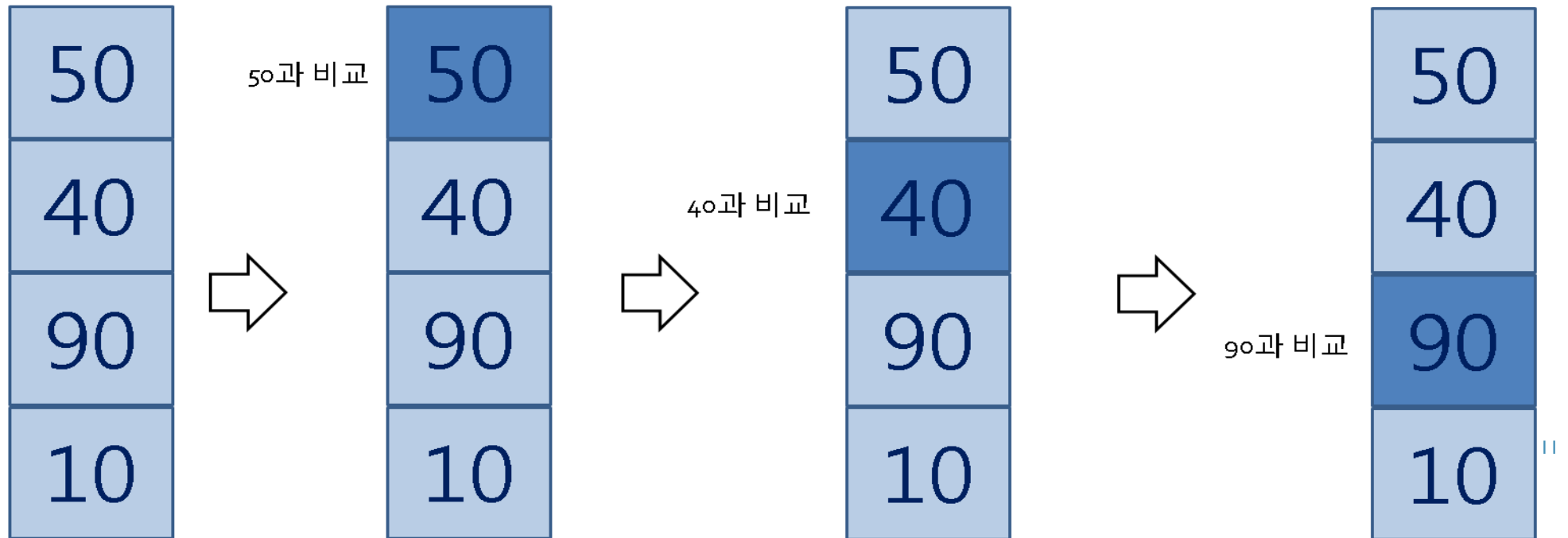


순차탐색(SEQUENTIAL SEARCH) 개념

- 기본 전략(오름차순)

- 원하는 값을 찾을 때까지 자료를 처음부터 끝까지 차례로 하나씩 비교하면서 찾음

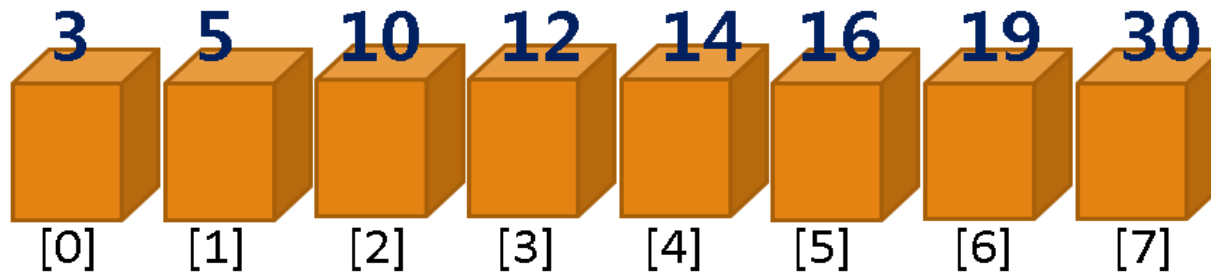
- 90찾기



순차탐색(SEQUENTIAL SEARCH) 개념

- 코딩

- data_list배열에 데이터가 저장되었을 때



[출력결과]

찾고 싶은 숫자를 입력하세요 : 14

순차탐색(SEQUENTIAL SEARCH) 개념

- 코딩

- data_list배열에 데이터가 저장되었을 때

[출력결과]

찾고 싶은 숫자를 입력하세요 : 14

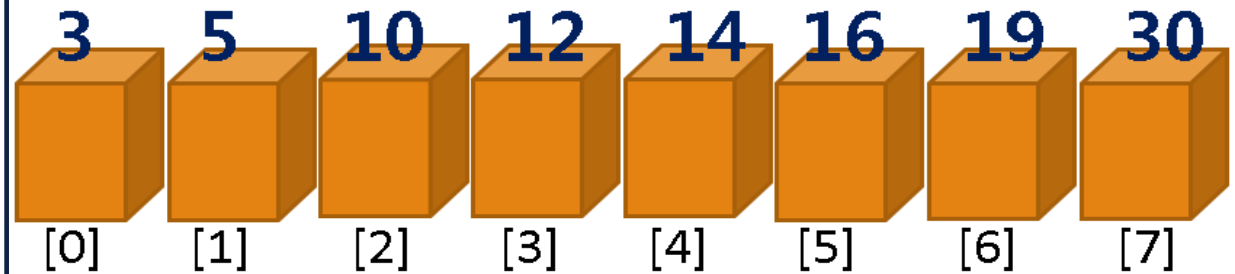
14와 0 번째의 값 3 비교
탐색 실패 !

14와 1 번째의 값 5 비교
탐색 실패 !

14와 2 번째의 값 10 비교
탐색 실패 !

14와 3 번째의 값 12 비교
탐색 실패 !

14와 4 번째의 값 14 비교
탐색 성공 탐색 위치 = 4



순차탐색(SEQUENTIAL SEARCH) 개념

■ 코딩

- data_list배열에 데이터가 저장되었을 때

[출력결과]

찾고 싶은 숫자를 입력하세요 : 14

14와 0 번째의 값 3 비교

탐색 실패 !

14와 1 번째의 값 5 비교

탐색 실패 !

14와 2 번째의 값 10 비교

탐색 실패 !

14와 3 번째의 값 12 비교

탐색 실패 !

14와 4 번째의 값 14 비교

탐색 성공 탐색 위치 = 4

index=0

target=0

data_list=[3, 5, 10, 12, 14, 16, 19, 30]

target =찾고 싶은 숫자를 입력하세요

while index < len(data_list):

print문으로 표현

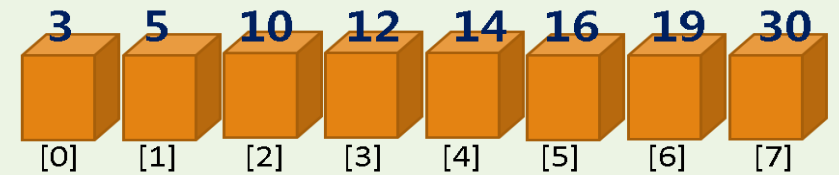
if target==data_list에 있는 데이터 값이 같으면 :

print("탐색 성공 탐색 위치 = ", data_list의 index 번호)

break

print(" 탐색 실패 ! ")

다음 을 검색하기 위해 index번호 증가



순차탐색(SEQUENTIAL SEARCH) 개념

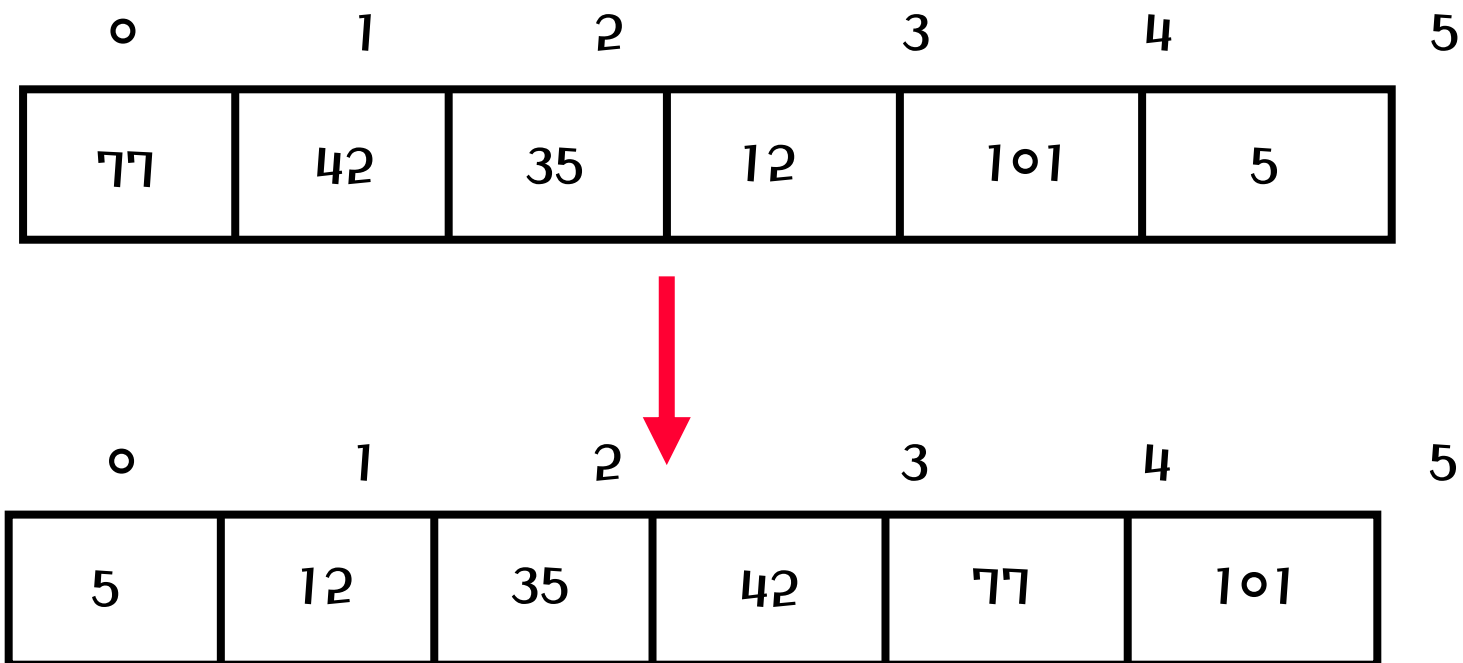
■ 순차탐색의 장단점

장점	<ul style="list-style-type: none">• 알고리즘이 단순하고 이해가 쉽다• 자료의 개수가 적을 때 좋다.• 자료가 정렬되어 있지 않아도 된다.
단점	<ul style="list-style-type: none">• 자료의 개수가 많아지면 비교 횟수가 많아진다.• 다른 방법에 비해 속도가 느리다.

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예



- 정렬 안된 수들의 집합의 값을 순서대로 정렬하기



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

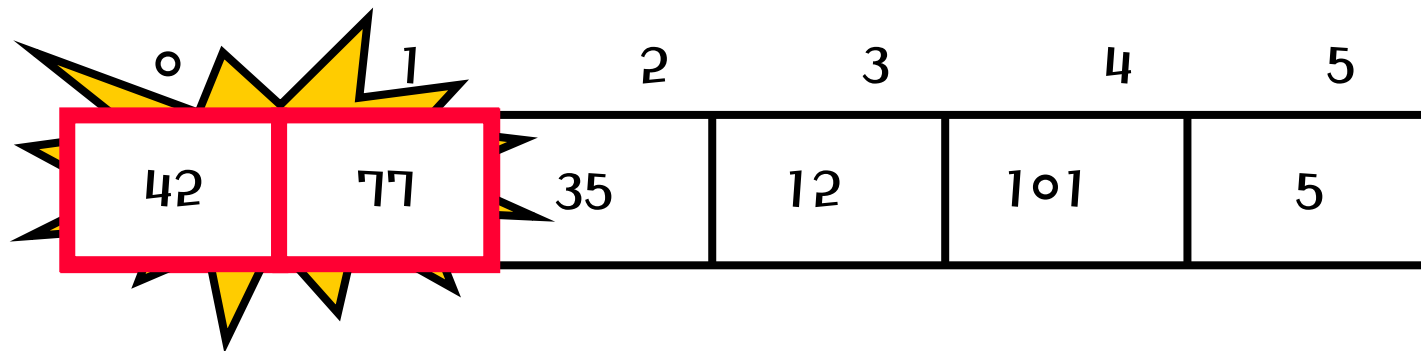
- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 두 개씩 값을 비교하며 큰 값이 맨 뒤에 위치하도록 반복하여 교체 (swap)

0	1	2	3	4	5
77	42	35	12	101	5

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

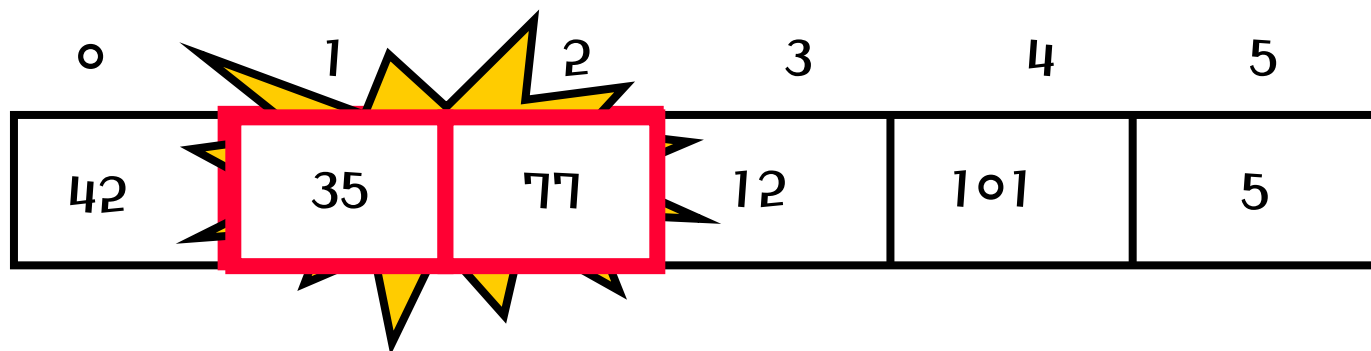
- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 맨 앞의 두 개 값을 비교하여 큰 값이 뒤에 위치하도록 교체 (swap)



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

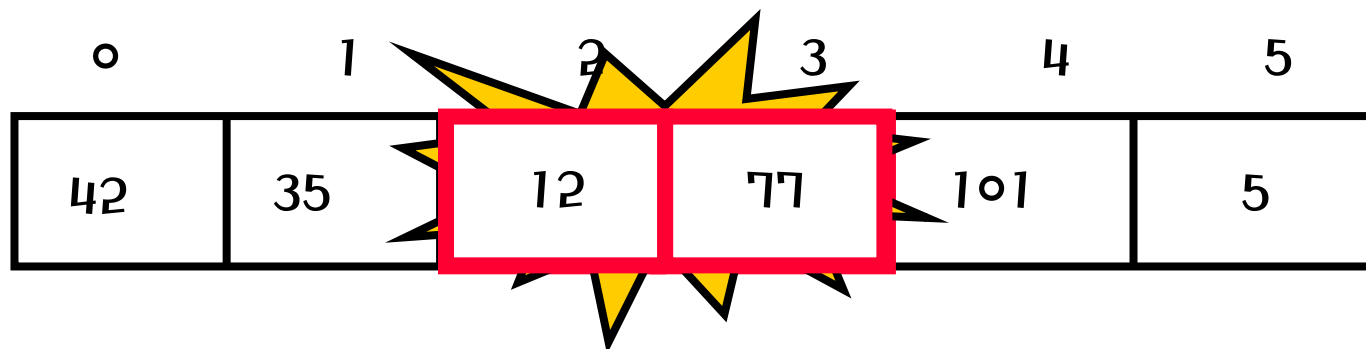
- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 두번째와 세번째의 두 개 값을 비교하여 큰 값이 뒤에 위치하도록 교체 (swap)



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 세번째와 네번째의 두 개 값을 비교하여 큰 값이 뒤에 위치하도록 교체 (swap)



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 네번째와 다섯번째의 두 개 값을 비교하여 큰 값이 뒤에 위치하도록 교체 (swap)

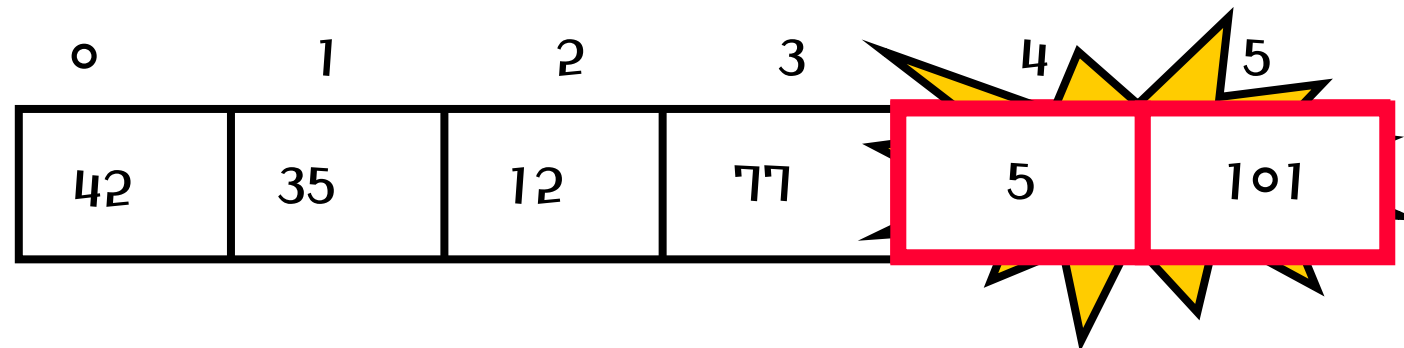
0	1	2	3	4	5
42	35	12	77	101	5

교체 발생하지 않음

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)

- 앞에서부터 하나씩 검토하며 뒤로 이동
- 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
- 다섯번째와 여섯번째의 두 개 값을 비교하여 큰 값이 뒤에 위치하도록 교체 (swap)



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬 (bubble sort)
 - 앞에서부터 하나씩 검토하며 뒤로 이동
 - 가장 큰 값을 “버블”과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
 - 여섯개의 값 중에서 마지막 6번째 값은 정렬된 위치에 저장 완료

0	1	2	3	4	5
42	35	12	77	5	101

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

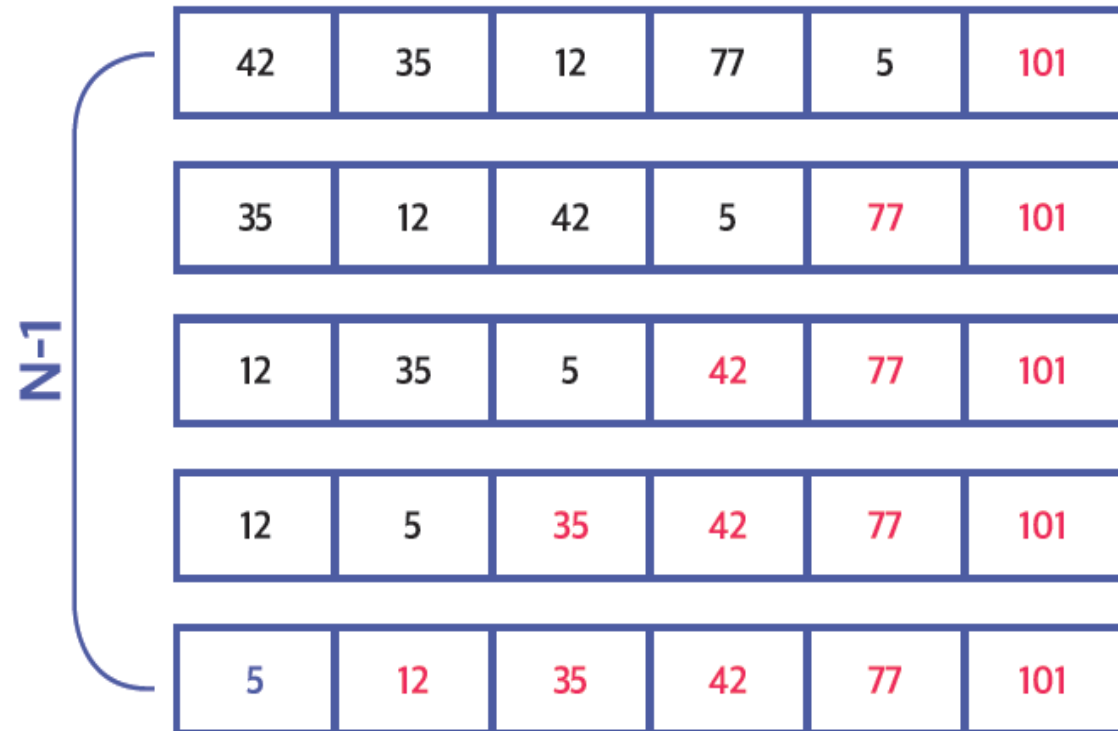
- 버블 정렬 (bubble sort)
 - 앞에서부터 하나씩 검토하며 뒤로 이동
 - 가장 큰 값을 "버블"과 같이 단계 별로 이동하며 가장 끝에 위치 시킴
 - 나머지 1-5개의 값에 대하여 반복 작업

0	1	2	3	4	5
5	12	35	42	77	101

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 버블 정렬(Bubble Sort)

[77, 42, 35, 12, 101, 5] 반복 횟수



단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 순차탐색(Sequential Search)
 - 처음부터 끝까지 하나하나 차례대로 비교하며 찾고자 하는 값을 탐색하는 방식
 - 자료가 정렬되지 않은 상태에서 특정 자료의 값을 찾는 경우에 적절
 - 자료가 존재한다면 자료의 위치 번호(index 값)를 반환
 - 자료가 존재하지 않는다면 탐색에 실패한 경우로, 일반적으로 -1을 반환

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘의 예

- 【예】 주어진 자료에서 46 탐색

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

index 0 3과 46은 같지 않음

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

생략 ...

index 10 46과 46은 같음 : 탐색 성공

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

단순하게 문제 풀기 (BRUTE FORCE) 방식 알고리즘 평가

- 장점
 - 다른 방법이 없을 때는 유일한 해결 방법
 - 모든 문제의 해결책으로 적용 가능
 - 모든 경우의 수를 모두 다 검토하고자 할 때 적절함
 - 문제의 구조화가 단순함
- 단점
 - 효율성이 보장되지 않음
 - 검토해야 할 대상의 범위가 광범위하면 적용하기가 적절하지 못함
 - 다른 문제 해결 방법보다 창의력이 떨어짐
 - 수행 시간이 받아들이기 힘들 정도로 길 수 있음

자료 탐색 기법_순차탐색

- 탐색의 개념
- 순차탐색의 알고리즘
- 버블정렬과 순차 탐색

