[슬라이드 1]

안녕하세요 이번에 C언어 정렬 알고리즘 세미나를 맡은 로보틱스 25기 김동현입니다

[슬라이드 2]

이번 세미나에서는 알고리즘에 대한 간단한설명과 버블정렬, 선택정렬, 삽입정렬을 배우고 직접 코딩실습을 해보는 시간을 가지겠습니다.

세미나 진행 도중에 문제가 있거나 궁금한 점이 있으면 바로 바로 손을 들고 말씀해주세요, 물론 끝나고 나서 질문도 받습니다.

[슬라이드 3]

저번 신재철 회원님의 포인터 세미나로 여러분들은 C언어의 기본문법을 모두 배웠습니다, 이제 이 문법들을 활용해서 알고리즘이란 것을 만들어 볼겁니다.

근데 알고리즘이 뭘까요?

'어떤 문제를 해결하기 위한 여러 동작들의 모임' 이게 알고리즘의 정의입니다.

지금 설명하는 부분은 모두 외우려고 할 필요는 없고 가벼운 마음으로 들으시면 됩니다.

알고리즘에는 기본적으로 5가지 구성요소가 있습니다. 0개이상의 외부입력, 1개이상의 출력, 모호하지 않고 명확한 구성, 주어진 시간 내에 종료 해야 하는 유한성, 명백하고 효율적으로 실행이 가능한 효율성 좀더 이해가 잘되도록

[슬라이드 4]

여러분의 학기초의 기억을 떠올리면서 복습하나 해볼까요? 로보틱스라는 글자를 10번 출력하는 프로그램을 프로그램을 머리속으로 한 번 상상해봅시다.

만약 여러분이 printf하나만을 알고있다면 저렇게 프로그램이 나오겠죠?

여기서 이걸 간단하게 만들려면 어떻게 해야 할까요? 대답해 보실분?

그럼 이게 여러분들이 만든 출력 알고리즘이에요. 0번의 입력, 10번의 출력, 명확한 구성, 유한성, 효율성을 모두 갖춘 알고리즘입니다.

알고리즘 별거 아니죠?

[슬라이드 5]

그럼 좀더 나아가서 알고리즘 복잡도에 대해서 알아보겠습니다.

알고리즘 복잡도는 크게 시간복잡도, 공간복잡도로 나눌 수 있습니다.

먼저 시간복잡도는 같은 프로그램에서 수행되는 시간이 더 짧을수록 좋은 알고리즘으로 알고리즘 패턴을 통해 대략적인 시간을 고려해서 Big O notation이라고 O옆에 괄호에 알고리즘이 수행되는데 걸리는

대략적인 시간을 표시합니다.

그 다음 같은 프로그램이라도 사용하는 공간이 적을수록 공간 낭비를 최소화 하여 메모리를 절약할수록 좋은 알고리즘이라고 할 수 있습니다.

오늘 공부할 정렬알고리즘에서는 공간복잡도는 크게 상관이 없으므로 넘어가겠습니다.

[슬라이드 6]

그럼 이제부터 집중해서 들어주세요. 그렇다면 우리가 배우는 정렬 알고리즘은 도대체 무엇을 하는 알고리즘일까요?

여기 앞에 무작위로 있는 데이터를 다음과 같이 크기 순으로 줄을 세우는 알고리즘을 정렬 알고리즘이라고 합니다.

[슬라이드 7]

그래서 오늘 배울 정렬 알고리즘은 아까 시간복잡도에서 배운 O(n^2)의 속도를 가진 알고리즘들을 배우겠습니다.

그 외에도 O(n log_2 n)속도를 가진 정렬 알고리즘도 있지만 이를 사용하기 위해서는 재귀함수에 대해서 알아야 하므로 궁금하신분은 개인적으로 찾아오시면 알려드리겠습니다.

그럼 오늘은 여기에 있는 순서대로 버블정렬, 선택정렬, 삽입정렬 순으로 세미나를 하겠습니다.

[슬라이드 8]

앞에 화면을 보면 a라는 5칸짜리 배열에 무작위의 숫자들이 들어있습니다.

정렬의 최종목표는 이 앞의 숫자들을 오름차순 혹은 내림차순으로 정렬시켜야 합니다.

오늘 세미나에서는 오름차순 기준으로만 설명을 진행하도록 하겠습니다.

버블정렬은 인접한 두 수를 비교해서 큰 수를 뒤로 보내는 알고리즘입니다.

직접 정렬이 되는 과정을 보여드리겠습니다.

먼저 첫번째 숫자를 index로 지정한다면 index는 0일테니, index+1번째는 1번째가 되겠죠?

이때 첫번째와 두번째 즉 index와 index+1번째의 자료를 비교하여줍니다.

오름차순 정렬을 한다면 작은숫자가 앞으로 와야하니 if문을 이용하여 앞의 숫자가 뒤의 숫자보다 더 큰지 검사를 하고

만약 앞의 숫자가 뒤의 숫자보다 더 크다면 교체를 해줍니다.

여기서는 서로 자리를 바꾸는 함수를 swap이라는 함수로 표현했지만 실제로 정의된 함수는 아니고 다음과 같이 여러분이 직접 구현하거나

main문에서 임시변수를 만들어서 처리해주어야 합니다.

여튼 index 즉 0번째 자리에는 90, index+1 자리에는 5가 있으므로 각 자리의 데이터를 교체해줍니다

교환을 완료하면 index는 1, index+1는 2가 되도록 index를 1증가시켜줍니다.

index가 증가되었으니 다시 반복하여 검사를 하면, index번째 즉 1번째에는 90, index+1번째 즉 2번째 자리에는 80이 있으므로

둘의 데이터를 교환합니다.

또 다시 index를 증가시켜 2번째와 3번째를 검사, 교환 3번째와 4번째를 검사 교환을 반복하여 index가 배열의 마지막에 도달할때까지 반복을 해 줍니다.

이렇게 index가 마지막에 도달하면 마지막번째 즉 4번째 데이터는 정렬이 완료가 된 상태가 됩니다.

이제 index를 다시 0으로 만들고 위의 과정을 반복해줍니다

0번째와 1번째를 검사하니 0번째 데이터가 작은 상태이므로 교환을 하지 않고 바로 index를 증가시켜 1번째와 2번째 데이터를 검사합니다.

1번째와 2번째는 1번째가 더 큰 상태이므로 교환하고 2번째와 3번째를 검사해서 교환하여 index가 기존에 정렬이 완료된 번째의 -1번째

즉 4번째 데이터가 보라색으로 정렬이 완료되었으니 -1시킨 3번째에 index가 도달하면 정렬이 완료가 된겁니다

또 이과정을 반복하여

0번째 1번째, 1번째 2번째

다시 index를 0으로 만들어 0번째와 1번째 이와 같은식으로 반복하면 정렬이 완료가 됩니다.

여기까지 이해가 안되거나 질문 있으신 분은 해주시기바랍니다.

[슬라이드 32]

다음은 선택정렬입니다. 선택정렬은 index를 기준으로 최소값을 찾고, 최소값과 index를 교환해주는 정렬입니다.

아까와 똑같은 데이터를 이번에는 선택정렬을 이용하여 오름차순 정렬을 해보겠습니다.

먼저 가장 앞에 있는 데이터를 index로 잡고, 화살표 범위를 화살표 방향으로 탐색하며 최소값을 찾습니다.

이 상태에서 최소값을 찾으면 5가 최소값으로 나옵니다. 그렇다면 최소값을 index와 교환하여줍니다.

교환을 하면, 기존에 index의 자리는 정렬이 완료된 데이터가 오게 됩니다.

이제 index에 1을 증가시켜 1번째 데이터를 index로 만들어줍니다.

또 다시 index를 기준으로 최소값을 찾으면 7이 최소값이 됩니다.

아까와 마찬가지로 index와 최소값을 교환해주면, index였던 1번째에도 정렬이 완료된 데이터가 오게 됩니다.

또 index를 1을 증가시키면, 2번째 데이터가 index가 되고 최소값 43과 교환

마찬가지로 index를 증가시켜 3번째 데이터가 index, 최소값도 80으로 교환을 하지만 자기 자신과의 교환이므로 실제로 변화는 생기지 않습니다.

최종적으로 index가 마지막에 도착하면 정렬이 완료가 됩니다.

여기까지 선택정렬에 대해서 이해가 안가거나 질문 있으신 분은 해주시기 바랍니다.

[슬라이드 44]

마지막으로 삽입정렬은 index로 지정된 데이터를 알맞은 위치에 삽입하는 정렬입니다.

삽입정렬의 index는 0번째가 아닌 첫 번째부터 시작하여 진행하게 됩니다. 탐색을 하기 전, index의 데이터를 임시로 만들어둔 변수에 저장을 해줍니다. 여기에 분홍색박스가 임시로 만든 변수입니다.

그렇다면 이제 빈 공간을 오른쪽으로 당기되, 두 가지 조건을 지키면서 당겨야 합니다.

첫째로 분홍색 임시변수에 들어있는 숫자보다 옮기어지는 값이 작아서는 안되고,

둘째로 배열의 0번째까지 도달하게 되면 더 이상 옮길 데이터가 없으므로 멈춰야 합니다.

이 두 가지 조건을 지키면서 정렬해보도록 하겠습니다.

아직 index 앞에 당길 데이터가 있고, 90은 5보다 크므로 기존에 5가 있던 위치에 90을 옮겨줍니다

이제 더 이상 옮길 데이터가 없으므로 반복을 멈추고, 임시변수에 저장해두었던 5를 a[0]에 넣어줍니다.

여기까지 정렬이 완료되었으므로 이제 다시 index를 1증가시켜 정렬을 시작합니다.

80을 임시변수에 저장하면, 앞에 옮길 데이터가 있고, 90은 80보다 크므로 기존에 80이 있던 자리에 90을 옮겨줍니다.

옮길 데이터가 있지만, 5는 80보다 작으므로 반복을 멈추고 임시변수에 들어있던 80을 a[1]자리에 넣어줍니다.

계속 반복해서 index를 1 증가시키고 43을 임시변수에 넣고

옮겨줄 데이터가 존재하고, 90은 43보다 크므로 이동, 80도 43보다 크므로 이동, 하지만 5는 43보다 작으므로 반복을 멈추고

43을 a[1]자리에 넣어줍니다.

또 반복해서 index를 1 증가시키고, 7을 임시변수에 넣고, 다시 쭉쭉쭉쭉 이동하여 옮겨주면 정렬이 완료됩니다.

이렇게 여러분은 버블 선택 삽입 3가지 정렬 알고리즘을 배우게 되었습니다.

이런 정렬 알고리즘은 기본적이고, 데이터를 효율적으로 처리하는 데에 있어서 필수적인 알고리즘입니다.

하도 중요해서 visual studio 라이브러리 안에도 있을 정도로 중요해요

가 아니라 지금은 여러분이 visual studio환경에서 편하게 배우면서 작업하지만, 여러분이 살아가면서 어떤 컴파일러를 마주하게 될지 모릅니다. 오늘 배운 정렬알고리즘 3가지를 모두 기억하는 것은 힘들겠지만 적어도 한가지방법은 기억해줬으면 좋겠다 가 아니라 기억해야 합니다. 다짜고짜 자는데 깨워서 정렬하라고 하면 할 수 있을 정도로 숙련을 해놓으면 좋겠네요

질문하실분 있습니까?

그럼 이제부터 오늘 배운 정렬 알고리즘 중 한 가지를 골라 직접 만들어보는 실습을 하겠습니다.