**Best of Best 11**

**Algorithms**

**김종현 멘토님 과제 #2 번**

**Best of Best 11기**

**보안제품 개발 희망 정경재**

**제출날짜:2022-07-08**

이번 BBC “The Secret Rules of Modern Living Algorithms”을 보며 느낀점은 알고리즘의 등장과 과거부터 현재까지의 알고리즘의 발전 방향과 미래에 어떠한 방향의 알고리즘이 나올지 생각해 볼 수 있었던 좋은 경험을 가질 수 있었던 시간이라 생각됩니다.

영상에서 가장 첫번째로 나온 알고리즘은 얼굴인식 알고리즘이지만 과거의 알고리즘부터 현재의 알고리즘까지 오는 관점을 설명하기 위해 뒤에서 말하는 것이 옳다고 생각해 해당 알고리즘은 나중에 말하고자 합니다.  
따라서 제 생각에 가장 알고리즘의 기본이 된다고 생각되는 알고리즘은 유클리드의 GCD알고리즘이라 생각합니다. 해당 알고리즘은 입력 받은 두 인수에 대해 최대공약수를 돌려주는 간단한 알고리즘입니다. 해당 알고리즘을 바탕으로 많은 사람들이 비교적 적은 노력으로 최대공약수를 구할 수 있었습니다.

다음으로는 google의 page rank알고리즘입니다. 대 인터넷 시대가 열리면서 사람들은 인터넷을 통해 정보들을 공유하게 되었고 따라서 정보의 홍수속에 빠지게 되었습니다. 이로 인해 다양한 정보들이 있지만 이 정보들 중 가장 신뢰도가 높은 정보를 쉽게 찾을 수 없었다는 것입니다. 레리페이지가 이러한 점에 착안해 page rank알고리즘을 구성하게 되며 각 페이지가 얼마나 많이 인용되었는지에 따라 웹에 weight를 적절하게 줘 이러한 정보의 정확성에 대한 문제점을 해결했다고 볼 수 있습니다.

다음은 정렬 알고리즘입니다.  
정렬알고리즘은 무질서한 순서를 지닌 순서를 질서 있고 원하는 순서를 지닌 순서로 변경하는 것을 의미합니다. 따라서 bubble sort, quick sort, insert sort, merge sort 등 다양한 정렬 방법 구체화한 정렬알고리즘이 있습니다. 따라서 각 알고리즘들은 각각의 장 단점을 지니고 있습니다.  
예를 들어 bubble의 경우는 극단적인 시간 복잡도를 지니지만 구현의 간단함을 지닌다는 장점을, merge sort의 경우 nlogn이라는 시간 복잡도를 지니지만 구현에 어려울 수 있다는 단점들을 가지고 있습니다. 이러한 정렬알고리즘을 통해 정방향 정렬을 통한 이진 탐색 등 다양한 정렬을 통해 쉽게 해결할 수 있게 문제를 만들어주었습니다.

다음은 매칭 알고리즘입니다.  
해당 매칭 알고리즘은 남녀가 어떻게 만나야 최적의 짝이 될 수 있는가에 대해서 부터 비롯된 알고리즘입니다.  
해당 알고리즘은 남녀문제를 해결한 공로로 인해 노벨상을 수상하게 되었습니다.

여기까지의 알고리즘은 상징주의적 발상으로 생성되었다고 이야기할 수 있을 것 같습니다.

지금까지 설명한 알고리즘들은 모두 알고리즘의 핵심인 로직을 어떻게 설계하고 구체화하는지에 대해 집중적이었기 때문이었습니다.

하지만 시대가 변하며 이러한 상징주의적인 관점으로는 완벽하게 문제를 풀 수 없게 되었고 이로 인해 연결 주의적인 관점이 시작하게 되었다고 생각합니다.

지금부터 설명할 알고리즘은 체스, 벌들의 비행 경로, Heathrow sequence algorithm, 얼굴인식 등과 같이 해당 문제들은 기존의 방식으로 Perfect한 정답을 제공하지는 못합니다.  
이유는 각 조건과 실시간적으로 상황이 변하지만 if-else같은 분기문과 반복문만으로는 이러한 문제들을 제대로 해결할 수 없기 때문입니다.

따라서 데이터를 통해 학습을 하는 연결주의가 대두 되었고 따라서 인공지능에서 ML을 통한 방법으로 문제를 완벽하진 않지만 효율적인 알고리즘을 배출하게 되었습니다.

Deep learning과 Game이론을 집합해 체스 알고리즘이 나오게 되었고 이와 같은 방식으로, 여러 경로에서 학습을 통해 최적의 거리를 이동하는 Traveling sales algorithm과 벌들의 비행 알고리즘, 항공에서 비행기의 이착륙을 조절하는 알고리즘이 나오게 되었습니다.

이후 다양한 필터들을 통해 특징들을 빠르게 인식해 얼굴인식을 하는 알고리즘, 물류의 최적의 경로를 제작하는 routing algorithm이 나오게 되었습니다.

이러한 알고리즘의 방향으로 볼 때 알고리즘은 현실의 문제를 해결을 하는 것을 목표로 하며 이전과는 다르게 완벽하게 현실의 문제를 푸는 알고리즘이 아니라 최선의 답을 낼 수 있는 연결주의적 기법의 알고리즘이 발전할 것으로 생각 되는 것을 느낄 수 있었던 영상이라 생각합니다.

이번에 제가 설명할 알고리즘은 정렬 알고리즘입니다.

정렬알고리즘은 다양한 알고리즘(Quick, bubble, merge, insert, ..)을 가지고 있습니다.

이중 Quick, bubble, insert, merge 정렬에 대해 이야기하고자합니다.

1. Quick 정렬  
   Quick 정렬을 Divide and Conque의 알고리즘 중 하나로 평균적으로 빠른 수행 속도를 자랑합니다.  
   해당 알고리즘의 동작 방법은 일단 첫번째 Pivot을 설정합니다.  
   해당 pivot을 기준으로 왼쪽은 작은 값, 오른쪽은 큰 값으로 정렬을 합니다.  
   첫번째 pivot을 기준으로 왼쪽에 새로운 pivot을 생성해 또 다시 이전의 작업을 반복하며 왼쪽을 정렬하고 오른쪽 역시 해당 방법을 그대로 작업해 정렬을 진행합니다.  
   해당 작업이 마무리 된다면 우리는 오름차순으로 정렬된 값을 얻을 것입니다.  
   가장 최악의 경우 n^2의 시간복잡도를 가지는 정렬알고리즘입니다.
2. Bubble 정렬  
   Bubble의 정렬은 가장 단순 무식한 방법이라 생각합니다.  
   말 그대로 비누방울이 올라가듯 첫번째 값부터 값을 비교하며 올라가며 정렬하는 방법입니다.  
   따라서 해당 알고리즘은 시작값과 다음 값의 값을 비교해 다음의 값이 이전 값보다 작다면 자리를 변경하고 그렇지 않으면 다음으로 넘어가는 작업을 모든 변수의 횟수에 대해 작업합니다.  
   각 변수에 대해 작업을 진행하므로 n^2의 시간복잡도를 가지는 정렬알고리즘입니다.
3. Insert 정렬  
   삽입 정렬의 기본적인 아이디어는 정렬하고자 하는 배열의 인덱스를 점차 늘려가는 것입니다.  
   삽입정렬의 시작은 주어진 배열의 첫번째 원소부터 시작합니다.  
   단일 원소로 있다는 것은 이미 정렬이 끝난 것이라 생각하면 됩니다.  
   따라서 배열의 크기를 하나 증가시키고 해당 배열 내에서 정렬을 시도합니다.  
   해당 과정들을 차근차근 시도하면 n^2의 시간복잡도를 지니는 정렬알고리즘이 완성됩니다.
4. Merge 정렬  
   이 알고리즘 역시 Divide & Conque의 알고리즘 중 하나로 가장 효율적인 알고리즘이라 생각됩니다.  
   일단 기본적인 아이디어는 각 원소들은 이미 정렬되었다는 것을 아이디어로 진행을 합니다.  
   각 원소들은 이미 정렬되어 있기에 인접한 이웃 원소들과 값을 비교해 새로운 배열들을 만듭니다.  
   이후 이 각각의 배열들은 이미 사전의 단계를 통해 정렬이 되있으므로 각각의 배열들의 원소들을 비교해서 새로운 정렬된 배열을 마지막까지 제작한다면 nlogn의 시간 복잡도를 가지는 정렬알고리즘이 완성됩니다.