

2주차 2차시 알고리즘의 설계와 분석 I

【학습목표】

1. 알고리즘의 설계방법을 이해할 수 있다.
2. 알고리즘을 분류하고 효율성을 이해할 수 있다.

학습내용1 : 알고리즘의 설계 방법

알고리즘의 설계방법은 쉽게 설명할 수 없는 창조적인 영역이다. 문제를 해결하는 사람의 창의적인 능력에 따라 다르지만 일반적인 알고리즘의 기법들은 그리디알고리즘, 동적 프로그래밍, 분할 정복 알고리즘, 무작위 알고리즘, 근사 알고리즘이 있다.

1. 그리디 알고리즘(Greedy)

그리디 알고리즘은 최적화 문제를 해결하는 알고리즘이다. 최적화문제는 가능한 해 중에서 가장 좋은 해를 찾는 문제이다. 그리디 알고리즘은 욕심쟁이방법, 탐욕적 방법, 탐욕 알고리즘이라 하기도 한다.

- ① 동전 거스름돈
- ② 최단 경로 찾기
- ③ 최소 신장 트리
- ④ 부분 배낭 문제
- ⑤ 집합 커버 문제
- ⑥ 작업 스케줄링

2. 동적 프로그래밍(Dynamic programming)

동적 프로그래밍알고리즘은 그리디 알고리즘과 같이 최적화문제를 해결하는 알고리즘이다. 동적 프로그래밍 알고리즘은 입력 크기가 작은 부분문제들을 모두 해결한 후에 그 해들을 이용하여 보다 큰 크기의 부분문제들을 해결하여 최종적으로 원래에 주어진 입력의 문제를 해결하는 알고리즘이다.

- ① 모든 쌍 최단 경로
- ② 편집 거리 문제
- ③ 배낭 문제
- ④ 연속 행렬 곱셈
- ⑤ 동전 거스름돈

3. 분할 정복(Divide-and-Conquer)알고리즘

분할 정복 알고리즘이란 주어진 문제의 입력을 분할 하여 문제를 해결하는 방식의 알고리즘이다. 분할된 입력에 대해서 동일한 알고리즘을 적용 해를 계산하며, 이들의 해를 취합하여 원래의 문제의 해를 구하는 알고리즘이다.

- 분할 : 정렬 하려는 배열을 $n/2$ 개의 부분배열로 분할한다.
- 정복 : 부분배열을 정렬 한다.
- 통합 : 정렬된 두 개의 배열을 합병 하여 하나의 정렬된 배열로 만든다.

- ① 합병 정렬
- ② 퀵 정렬
- ③ 선택 문제

학습내용2 : 알고리즘의 효율성 분석

알고리즘의 효율성은 알고리즘의 수행 시간 또는 알고리즘이 수행하는 동안 사용되는 메모리의 공간의 크기로 나타낼 수 있다. 이들을 각각 이들을 시간 복잡도(time complexity)와 공간 복잡도(space complexity)라 한다. 일반적인 알고리즘을 비교할 때는 시간복잡도가 주로 사용된다.

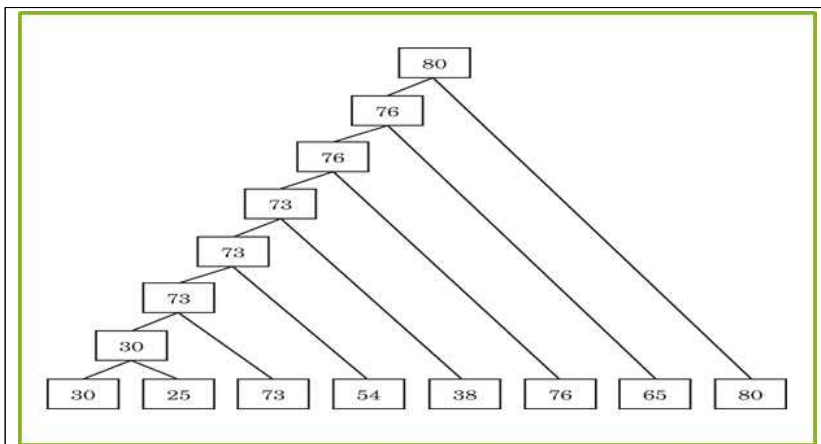
알고리즘을 프로그램으로 구현해서 이를 컴퓨터에 실행된 시간을 측정할 수도 있다. 그러나 실제로 측정된 시간은 알고리즘을 수행하는데 여러 가지 변수가 있으므로, 알고리즘의 효율성을 객관적으로 평가 하기 어렵다. 왜냐하면 어떤 컴퓨터에서 수행시켰는지, 어떤 프로그래밍 언어를 사용했는지, 얼마나 숙련된 프로그래머가 프로그램을 작성 했는지에 따라 실제 수행시간은 얼마든지 달라질 수 있다.

따라서 시간 복잡도는 알고리즘이 수행하는 기본적인 연산 횟수를 입력크기에 대한 함수로 표현한다.

예를 들어서 10장의 숫자 카드 중에서 최대 숫자를 찾는데 순차 탐색으로 찾는 경우 숫자의 비교가 기본연산이고, 총 비교 횟수는 9가된다. 만약 n 장의 카드가 있다면 $(n-1)$ 번의 비교를 수행하므로 $(n-1)$ 의 시간 복잡도가 된다.

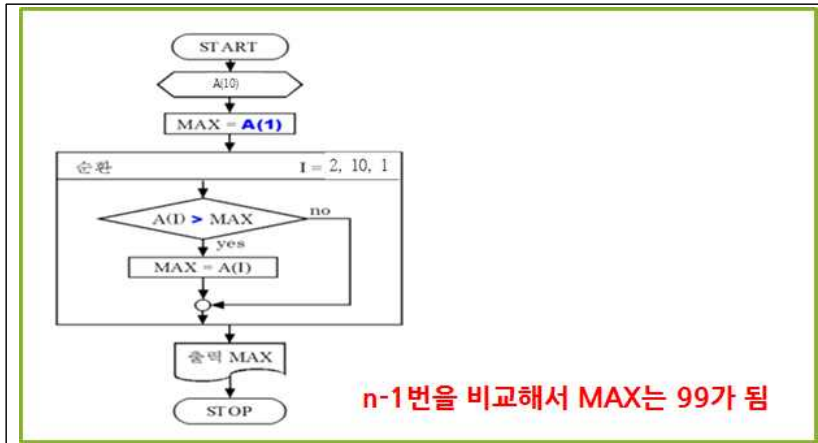
1. $(n-1)$ 번의 비교

* $(n-1)$ 번의 비교를 해서 최대값 구하는 예



2. 최대값 구하는 순서도

* 입력 data 25, 75, 65, 24, 87, 76, 90, 88, 99, 80일 때



3. 최대값 구하는 수도코드

알고리즘 GetLargest

입력: 숫자들의 리스트 L.

출력: 리스트에서 가장 큰 값

```

largest ← L[0]
for each n in L do
  if n > largest then
    largest ← n
return largest
  
```

4. 1에서 10까지 구하는 알고리즘 분석

① 1부터 10까지의 숫자를 직접 하나씩 더한다.

$$1 + 2 + 3 + \dots + 10 = 55$$

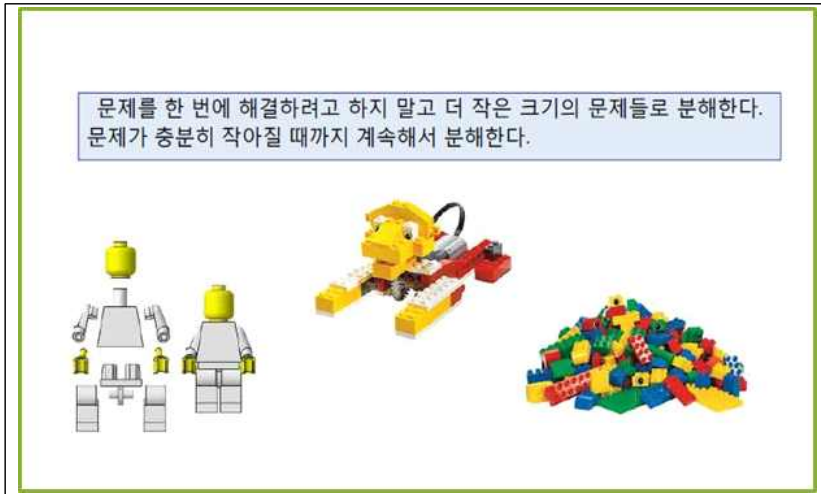
② 두수의 합이 10이 되도록 숫자들을 그룹핑하여 그룹의 개수에 10을 곱하고 남은 숫자 5를 더한다.

$$\begin{array}{l}
 (0 + 10) = 10 \\
 (1 + 9) = 10 \\
 (2 + 8) = 10 \\
 (3 + 7) = 10 \\
 (4 + 6) = 10 \\
 5
 \end{array}
 \quad \left\{ \begin{array}{l} 10 \times 5 = 50 \\ + 5 \\ = 55 \end{array} \right.$$

③ 공식을 이용하여 계산할 수도 있다.

$$10 * (1 + 10) / 2 = 55$$

5. 알고리즘을 만드는 방법의 예



6. 알고리즘의 분석 방법

* 알고리즘의 복잡도를 표현하는 데는 아래와 같은 분석 방법이 있다.

- ◎ 최악 경우 분석
- ◎ 평균 경우 분석
- ◎ 최선 경우 분석

일반적으로 최악 경우 분석으로 알고리즘의 복잡도를 나타낸다. 최악 경우 분석은 어떤 입력이 주어지더라도 얼마 이상을 넘지 않는다. 라는 표현이다.

평균 경우 분석은 입력의 확률 분포를 가정하여 분석하는데 대부분의 경우 균등 분포를 가정한다. 최선경우분석은 거의 사용되지 않으나 최적알고리즘을 고안하는데 참고 자료가 된다.

1) 각각 분석에 대한 예

① 최선 경우 분석

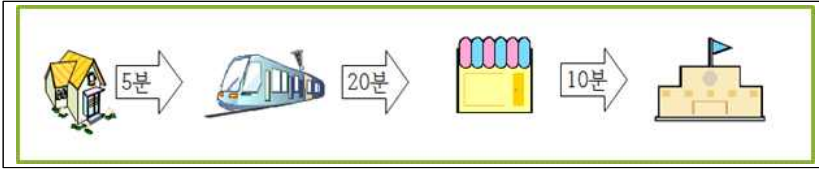
하준이는 매일 지하철을 타고 학교에 간다. 집에서 지하철까지는 5분이 걸리고 지하철을 타면 학교 근처 역까지 20분이 걸리며, 강의실까지는 10분이 걸린다고 가정하자.



이때 최선의 경우는 철수가 집을 나와서 5분후 지하철에 도착하고 바로 지하철을 탄 경우 최선의 경우 시간은 $5+20+10=35$ 분이다.

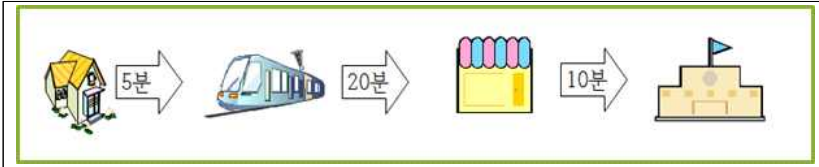
② 최악 경우 분석

최악의 경우는 지하철에 도착하여 승차하려는 순간 지하철이 출발하여 다음 지하철을 기다려야 하는 경우이다.



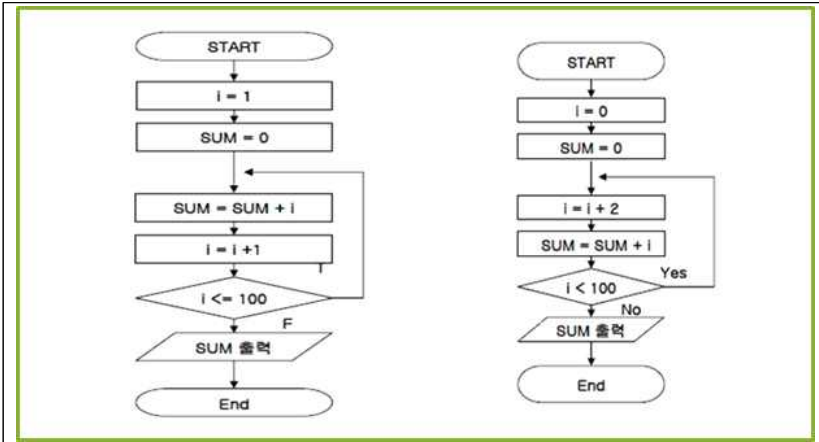
다음 지하철이 6분후에 온다고 도착한다고 하면 최악 경우 소요되는 시간은 $35+6=41$ 분이다.

③ 평균 경우 분석



평균 시간은 최악과 최선의 중간이라고 가정 하면 $35+3=38$ 분이 된다.

2) 예제 : 다음 순서도의 비교 회수를 알아 보자



【학습정리】

1. 알고리즘의 설계 방법

- 알고리즘의 설계방법은 쉽게 설명할 수 없는 창조적인 영역이다. 문제를 해결하는 사람의 창의적인 능력에 따라 다르지만 일반적인 알고리즘의 기법들은 그리디알고리즘, 동적 프로그래밍, 분할 정복 알고리즘, 무작위 알고리즘, 근사 알고리즘이 있다.

2. 알고리즘의 효율성 분석

- 알고리즘의 효율성은 알고리즘의 수행 시간 또는 알고리즘이 수행하는 동안 사용되는 메모리의 공간의 크기로 나타낼 수 있다. 이들을 각각 이들을 시간 복잡도(time complexity)와 공간 복잡도(space complexity)라 한다. 일반적인 알고리즘을 비교할 때는 시간복잡도가 주로 사용된다.

3. 알고리즘의 분석 방법

* 알고리즘의 복잡도를 표현하는 데는 아래와 같은 분석 방법이 있다.

- 최악 경우 분석
- 평균 경우 분석
- 최선 경우 분석

일반적으로 최악 경우 분석으로 알고리즘의 복잡도를 나타낸다. 최악 경우 분석은 어떤 입력이 주어져더라도 얼마 이상을 넘지 않는다. 라는 표현이다.

평균 경우 분석은 입력의 확률 분포를 가정하여 분석하는데 대부분의 경우 균등 분포를 가정한다. 최선경우분석은 거의 사용되지 않으나 최적알고리즘을 고안하는데 참고 자료가 된다.