

05강

알고리즘과 자료구조

분할 정복과 동적프로그래밍

서울과학기술대학교 신일훈 교수

학습목표

- 1 병합정렬을 수행하는 알고리즘을 설계할 수 있다.
- 2 피보나치 수열을 구하는 분할정복의 한계를 이해한다.
- 3 동적 프로그래밍의 개념을 이해한다.
- 4 동적 프로그래밍을 적용하여 피보나치 수열을 구하는 알고리즘을 설계할 수 있다.





병합 정렬

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

아이디어

```
list = [2, 5, 8, 3, 11, 7, 6, 9]
```

리스트를 두 개의 서브리스트로 분할하여 각각을 정렬 (더 작은 문제로 쪼갬)

각각의 정렬된 서브리스트를 병합

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

아이디어

```
list = [2, 5, 8, 3, 11, 7, 6, 9]
```

1. [2, 5, 8, 3], [11, 7, 6, 9]
2. [2, 5], [8, 3], [11, 7, 6, 9]
3. [2], [5], [8, 3], [11, 7, 6, 9]
4. [5, 2], [8, 3], [11, 7, 6, 9]
5. [5, 2], [8], [3], [11, 7, 6, 9]

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

아이디어

5. [5, 2], [8], [3], [11, 7, 6, 9]
6. [5, 2], [8, 3], [11, 7, 6, 9]
7. [8, 5, 3, 2], [11, 7, 6, 9]
8. ...
9. [8, 5, 3, 2], [11, 9, 7, 6]
10. [11, 9, 8, 7, 6, 5, 3, 2]

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

의사코드

```
def merge_sort(list) :  
    length = len(list)  
    if (length == 1) :  
        return list  
  
    start1 = 0  
    start2 = length // 2
```

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

의사코드

```
def merge_sort(list) :  
    length = len(list)  
    if (length == 1) :  
        return list  
  
    start1 = 0  
    start2 = length // 2
```

```
list1 = list[start1:start2]  
list2 = list[start2:length]  
list1 = merge_sort(list1)  
list2 = merge_sort(list2)  
result = merge(list1, list2)  
return result
```


1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

병합 아이디어

1. [8, 5, 3, 2] , [11, 9, 7, 6] => []
2. [8, 5, 3, 2] , [9, 7, 6] => [11]
3. [8, 5, 3, 2] , [7, 6] => [11, 9]
4. [5, 3, 2] , [7, 6] => [11, 9, 8]
5. [5, 3, 2] , [6] => [11, 9, 8, 7]
6. [5, 3, 2] , [] => [11, 9, 8, 7, 6]
7. [], [] => [11, 9, 8, 7, 6, 5, 3, 2]

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

의사코드

```
def merge(list1, list2) :  
    result = []  
    while (len(list1) > 0 and  
           len(list2) > 0) :  
        if(list1[0] > list2[0]) :  
            num = list1.pop(0)  
            result.append(num)  
        else :  
            result.append(list2.pop(0))  
    while (len(list1) > 0) :  
        result.append(list1.pop(0))  
    while (len(list2) > 0) :  
        result.append(list2.pop(0))  
    return result
```

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

의사코드

```
def merge_sort(list) :  
    ...  
    list1 = list[start1:end1]  
    list2 = list[start2:end2]  
    list1 = merge_sort(list1)  
    list2 = merge_sort(list2)  
    result = merge(list1, list2)  
    return result
```

최악 시간복잡도 ?

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

아이디어

```
list = [2, 5, 8, 3, 11, 7, 6, 9]
```

1. [2, 5, 8, 3], [11, 7, 6, 9]
2. [2, 5], [8, 3], [11, 7], [6, 9]
3. [2], [5], [8], [3], [11], [7], [6], [9]
4. [5, 2], [8, 3], [11, 7], [9, 6]
5. [8, 5, 3, 2], [11, 9, 7, 6]
6. [11, 9, 8, 7, 6, 5, 3, 2]

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

의사코드

```
def merge_sort(list) :  
    ...  
    list1 = list[start1:end1]  
    list2 = list[start2:end2]  
    list1 = merge_sort(list1)  
    list2 = merge_sort(list2)  
    result = merge(list1, list2)  
    return result
```

최악 시간복잡도 : $O(N\log N)$

1. 병합(merge) 정렬

- N개의 숫자를 저장한 파이썬 리스트를 내림차순으로 정렬하시오.

파이썬 코드 실행

```
list = [3, 5, 7, 1, 8, 2]  
print(merge_sort(list))
```

```
In [43]: runfile('D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/  
untitled1.py', wdir='D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')  
[8, 7, 5, 3, 2, 1]
```



동적 프로그래밍 개념

2. 동적 프로그래밍 개념

■ 피보나치 수열을 분할정복으로 해결할 때의 문제점

- 피보나치 수열을 구하는 과정 (재귀)

- $f(10) = f(9) + f(8)$
- $f(9) = f(8) + f(7)$
- $f(8) = f(7) + f(6)$
- ...

=> 동일한 부분 문제를 반복하여 풀고 있음을 알 수 있음.

- 가령, $f(9)$ 를 구할 때 $f(8)$ 의 답을 구하지만, $f(10)$ 을 계산할 때, $f(8)$ 을 다시 계산함

2. 동적 프로그래밍 개념

■ 피보나치 수열을 분할정복으로 해결할 때의 문제점

- 동일한 부분 문제를 반복하여 풀기 때문에 시간복잡도가 높음.
 - $O(\log 2^N)$
 - 이미 구한 $f(k)$ 를 메모리에 저장하고 추후 필요할 때마다 이를 활용하면, 시간복잡도를 크게 개선할 수 있음
- => dynamic programming 기법 활용

2. 동적 프로그래밍 개념

■ 피보나치 수열을 분할정복으로 해결할 때의 문제점

파이썬 코드

```
import time

def fibo(num) :
    if (num == 1 or num == 2) :
        return 1
    return (fibo(num - 1) + fibo(num - 2))
```

2. 동적 프로그래밍 개념

❑ 피보나치 수열을 분할정복으로 해결할 때의 문제점

파이썬 코드 실행

```
start = time.time()
res = fibo(20)
end = time.time()
print(res, end-start)
```

```
start = time.time()
res = fibo(30)
end = time.time()
print(res, end-start)
```

2. 동적 프로그래밍 개념

■ 피보나치 수열을 분할정복으로 해결할 때의 문제점

파이썬 코드 실행

```
start = time.time()
res = fibo(40)
end = time.time()
print(res, end-start)
```

```
In [56]: runfile('D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/
untitled1.py', wdir='D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')
6765 0.001995086669921875
832040 0.1316697597503662
102334155 17.545026063919067
```



동적 프로그래밍을 활용한 피보나치 수열

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

아이디어

- $\text{fibo}(n) = \text{fibo}(n-1) + \text{fibo}(n-2)$, $\text{fibo}(0) = 0$, $\text{fibo}(1) = 1$
- memoization?
 - 한번 수행한 연산 결과를 저장 => dictionary, list 등 활용

3. 동적 프로그래밍을 활용한 피보나치 수열

- ❑ 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드 - 리스트 활용

```
def fibo(num, list) :  
    if (num == 0 or num == 1) :  
        return num  
    if (list에 답이 저장되어 있다면) :  
        return list[num]  
    else :  
        list[num] = fibo(num-1, list) + fibo(num-2, list)  
        return list[num]
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드 - 리스트 활용

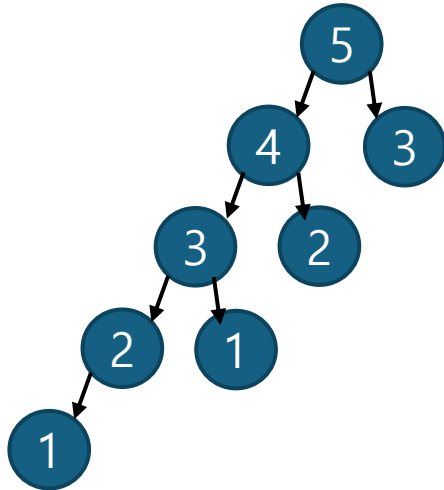
```
def fibo(num, list) :  
    if (num == 0 or num == 1) :  
        return num  
    if (list에 답이 저장되어 있다면) :  
        return list[num]  
    else :  
        list[num] = fibo(num-1, list) + fibo(num-2, list)  
        return list[num]
```

최악 시간복잡도?

3. 동적 프로그래밍을 활용한 피보나치 수열

- ❑ 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드 - 리스트 활용



3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드1 - 리스트 활용

```
def fibo(num, list) :  
    if (num == 0 or num == 1) :  
        return num  
    if (list에 답이 저장되어 있다면) :  
        return list[num]  
    else :  
        list[num] = fibo(num-1, list) + fibo(num-2, list)  
        return list[num]
```

최악 시간복잡도 :
 $O(N)$

3. 동적 프로그래밍을 활용한 피보나치 수열

- ❑ 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드1 - 리스트 활용

```
def fibo(num) :  
    list = []  
    for i in range(num+1):  
        list.append(-1)  
    return fibo_list(num, list)
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드 - 리스트 활용

```
def fibo_list(num, list):  
    if (num == 0 or num == 1):  
        return num  
    if (list[num] != -1):  
        return list[num]  
    else :  
        list[num] = fibo_list(num-1, list) + fibo_list(num-2, list)  
        return list[num]
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드 실행 - 리스트 활용

```
import time
start = time.time()
res = fibo(30)
end = time.time()
print(res, end-start)
start = time.time()
res = fibo(40)
end = time.time()
print(res, end-start)
```

```
In [57]: runfile('D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/
untitled1.py', wdir='D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')
832040 0.0
102334155 0.011968612670898438
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- ❑ 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드2 - 딕셔너리 활용

- 딕셔너리 : (key, value)의 집합
 - key를 통해 value를 찾을 수 있음.

3. 동적 프로그래밍을 활용한 피보나치 수열

- ❑ 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

의사코드2 - 딕셔너리 활용

```
def fibo(num, dict) :  
    if (num == 0 or num == 1) :  
        return num  
    if (num in dict) :  
        return dict[num]  
    else :  
        dict[num] = fibo(num-1, dict) + fibo(num-2, dict)  
        return dict[num]
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드2 - 딕셔너리 활용

```
def fibo(num) :  
    dict = {}  
    return fibo_dict(num, dict)
```


3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드2 - 딕셔너리 활용

```
def fibo_dict(num, dict) :  
    if (num == 0 or num == 1) :  
        return num  
    if (num in dict) :  
        return dict[num]  
    else :  
        dict[num] = fibo_dict(num-1, dict) + fibo_dict(num-2, dict)  
        return dict[num]
```

3. 동적 프로그래밍을 활용한 피보나치 수열

- 피보나치 수열을 동적 프로그래밍을 활용하여 구하시오.

파이썬 코드2 실행 - 디렉터리 활용

```
import time
start = time.time()
res = fibo(30)
end = time.time()
print(res, end-start)
start = time.time()
res = fibo(40)
end = time.time()
print(res, end-start)
```

```
In [60]: runfile('D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/
untitled1.py', wdir='D:/data/lecture/
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')
832040 0.0
102334155 0.0
```

정리하기

- ✓ 병합정렬을 수행하는 알고리즘
- ✓ 분할정복의 한계
- ✓ 동적 프로그래밍 개념
- ✓ 동적 프로그래밍을 활용한 피보나치 수열 알고리즘

06강

다음시간 안내▶▶▶

탐욕 알고리즘