

1. 10

```

1 int S = 0; ①
2 for (int i = 0; i < N; i++) 0부터 N까지 반복 (N+1)
3   for (int j = 0; j < N; j++) 2행 N번 x 3행 N+1번
4     S += N; 1회 x 2행 N번 x 3행 N번

```

$\therefore 1 + N + 1 + N(N+1) + N^2 \sim N^2$

$f(N) = 2N^2 + 2N + 2 \quad g(N) = N^2$

$C_1 g(N) \geq f(N) \geq C_2 g(N) \quad N=1 \quad 100 \geq 6 \geq 1$
 $C_1 = 100 \quad C_2 = 1$

$f(N) = \Theta(N^2)$

1. 11

```

1 int S = 0; ①
2 for (int i = 0; i < N; i++) 0부터 N까지 (N+1)
3   for (int j = 0; j < i; j++)
4     S += N;

```

$i=0 \quad j < 0 \quad 1$
 $i=1 \quad j < 1 \quad 2$
 $i=2 \quad j < 2 \quad 3$
 \vdots
 $i=N-1 \quad j < N-1 \quad N-1$

2행 (N+1)
 3행 1 ~ N-2 까지
 $\frac{(N-2)(N-1)}{2}$
 4행 1회

$\hookrightarrow \frac{(N-1)N}{2}$

$\therefore 1 + N + 1 + \frac{N(N-1)}{2} + \frac{(N-2)(N-1)}{2}$

$f(N) = N^2 - N + 3$
 $g(N) = N^2$

$C_1 g(N) \geq f(N) \geq C_2 g(N)$
 $C_1 = 100 \quad C_2 = 1$

$N=1 \quad 100 \geq 3 \geq 1$

$f(N) = \Theta(N^2)$

1.8

a) $10N^2 - 3N + 9 = f(N)$ $g(N) = N^2$

$$f(N) \leq C g(N)$$

$$C = 100 \quad N = 1$$

$$10 - 3 + 9 \leq 100 \cdot 1$$

$$\therefore f(N) = O(N^2)$$

b) $2N^2 + N \log N + 5N = f(N)$

$$g(N) = N^2$$

$$f(N) \leq C g(N)$$

$$C = 100 \quad N = 1$$

$$2 + 0 + 5 \leq 100 \cdot 1$$

$$\therefore f(N) = O(N^2)$$

c) $8N^3 + 3N + 5 = f(N)$ $g(N) = N^3$

$$f(N) \leq C g(N)$$

$$C = 100 \quad N = 1$$

$$8 + 3 + 5 \leq 100 \cdot 1$$

$$\therefore f(N) = O(N^3)$$

d) $2^N + N^3 + 5 = f(N)$

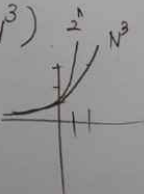
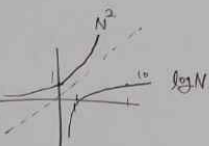
$$g(N) = 2^N$$

$$f(N) \leq C g(N)$$

$$N = 1 \quad C = 100$$

$$2^1 + 1 + 5 \leq 100 \cdot 2$$

$$\therefore f(N) = O(2^N)$$



1. 19 f(4)

```
public static void f(int N) {
    System.out.print(N);
    if (N > 0) f(N-1);
}
```

∴ 4321

- ① f(4) 호출
- ② 4 출력
- ③ 조건 만족 → f(3) 호출
- ④ f(3) 호출
- ⑤ 3 출력
- ⑥ 조건 만족 → f(2) 호출
- ⑦ f(2) 호출
- ⑧ 2 출력
- ⑨ 조건 만족 → f(1) 호출
- ⑩ f(1) 호출
- ⑪ 1 출력
- ⑫ 조건 만족 X → 종료

1. 20 g(4)

```
public static void g(int N) {
    if (N > 0) g(N-1);
    System.out.print(N);
}
```

∴ 0

- ① g(4) 호출
- ② 조건 만족 → g(3) 호출
- ③ g(3) 호출
- ④ 조건 만족 → g(2) 호출
- ⑤ g(2) 호출
- ⑥ 조건 만족 → g(1) 호출
- ⑦ g(1) 호출
- ⑧ 조건 만족 → g(0) 호출
- ⑨ g(0) 호출
- ⑩ 조건 만족 X → 종료

1. 21 h(4)

```
public static void h(int N) {
    System.out.print(N);
    if (N > 0) h(N-2);
    System.out.print(N);
}
```

h(4) → h(2) → h(0)

420024

- ① N 호출 → 4 출력
- ② h(4 > 0) h(2) 호출
- ③ 2 출력
- ④ h(2 > 0) h(0) 호출
- ⑤ 0 호출 → 0 출력
- ⑥ 조건 만족 X
- ⑦ 0 출력
- ⑧ 2 출력
- ⑨ 4 출력

4
2
0
0
2
4

1.22 h(4)

```
public static void h(int N) {
    if (N > 0) {
        h(N-1);
        System.out.print(N);
    }
}
```

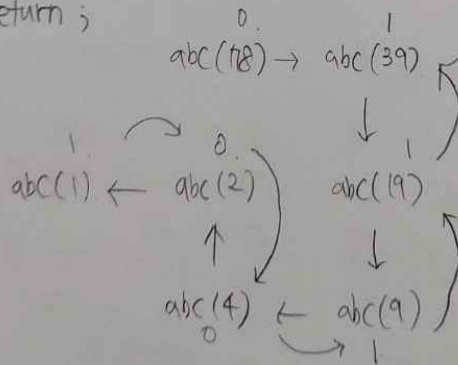
$h(4) \rightarrow h(3) \rightarrow h(2) \rightarrow h(1)$

$\therefore 1234$

- ① h(4)
- ② if(4>0)
- ③ h(3) → ① h(3)
- ④ if(3>0)
- ⑤ h(2) → ① h(2)
- ⑥ if(2>0)
- ⑦ h(1) → ① h(1)
- ⑧ if(1>0)
- ⑨ h(0) X

1.23 abc(78)

```
public static void abc(int N) {
    int r = N % 2;
    System.out.print("X");
    if (N >= 2) {
        abc(N/2);
        System.out.printf("%d", r);
    }
    return;
}
```



$\therefore \text{XXXXXXXX1001110}$

- ① 1 줄
- ② 2 줄
- ③ 3 줄
- ④ 4 줄

- ① abc(78)
- ② r=0 ... abc(78)
- ③ *
- ④ 78>=2 ⑤ abc(39)
- ⑥ r=1 ... abc(39)
- ⑦ *
- ⑧ 39>=2 ⑨ abc(19)
- ⑩ r=1 ... abc(19)
- ⑪ *
- ⑫ 19>=2 ⑬ abc(9)
- ⑭ r=1 abc(9) * abc(4)
- ⑮ r=0 abc(4) * abc(2)
- ⑯ r=0 abc(2) * abc(1)
- ⑰ r=1
- ⑱ *
- ⑲ if X
- ⑳ 1

1.24 test ("11010011", 4)

```
public static int test (String s, int last) {
    if (last < 0) { ①
        return 0;
    } if (s.charAt (last) == '0') { ②
        return 2 * test (s, last-1);
    }
    return 1 + 2 * test (s, last-1); ③
}
```

26
 ① X → ② 2 * test (s, 3); → ① X → ② X → ③
 1 + 2 * test (s, 2); → ① X → ② 2 * test (s, 1);
 → ① X → ② X → ③ 1 + 2 * test (s, 0) → ① X
 → ② X → ③ (return 1) + 2 * test (s, -1) → ① 0

1.25

```
public static void t (int N) {
    if (N > 0) {
        t (N/2);
        System.out.print (N%2);
    }
}
```

1. N 1011 → 2210121

t(100) → t(50) → t(25)
 → t(12) → t(6) → t(3)
 → t(1) → t(0)
 (1)

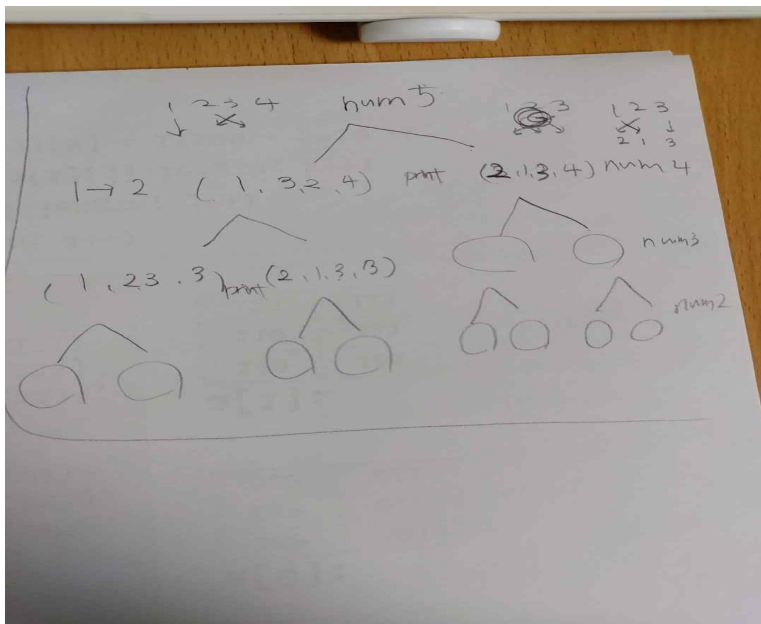
1100100

2 | 100
 2 | 50 0
 2 | 25 0
 2 | 12 1
 2 | 6 0
 2 | 3 0
 1 1

1100100

하노이탑 감상문

하노이탑 규칙이 큰원판이 작은원판위로 갈 수 없어서 직접원판을 움직이면서 어떻게 옮기는지 해봤던 기억이 있어서 와.. 이거를 어떻게 표현했을까 하며 소스코드를 보는데 생각지도 못한 방식으로 코드가 나오니까 정말 신기하게 보고 있습니다. 물론 이전에는 재귀함수에 대해서 잘 모르는 상태이기에 일반적인 소스코드로는 실행이 안됐던 기억이 납니다. 분명 군대에 있을 때 사지방에서 알고리즘을 공부하는 사이트에 있던 문제였던 것 같습니다. 어떻게 푸는지 궁금해서 몇일 동안 계속 고민하다가 문제의 정답을 보고 당시에만 해도 1줄도 이해하지 못했던 제가 지금와서보니 어느 정도 흐름이 보이는 것 같아서 군대에서 했던 그 공부들이 헛되지 않아서 다행이다 라는 생각이 듭니다. 하노이탑 소스코드를 보고 흐름도를 보니 2위치와 3위치에 있는 원판을 옮기는 문제와 1위치에 있는 원판을 3위치에 옮기는 문제로 나뉘서 모든 경우에 수를 실행하는 코드인 것 같습니다. 그렇게 흐름도를 보면은 전체 모습이 2진 트리? 라고 1학년 때 이산수학에서 봤던 것 같은 모양이 나와서 해당하는 점 위치가 하노이 함수를 사용한 횟수이지 않을까 생각이 듭니다. 이상이 제가 하노이탑 소스코드를 보고 생각한 느낀점입니다. 아래는 흐름도(?)입니다.



하노이탑 실행 결과

```

3
1 : 1 -> 3
2 : 1 -> 2
1 : 3 -> 2
3 : 1 -> 3
1 : 2 -> 1
2 : 2 -> 3
1 : 1 -> 3
    
```

피보나치 함수

시간 비교

속도비교	
recursive	
10/2.454초	정상
20/2.212초	정상
30/3.657초	정상
40/3.588초	정상
50/66.668초	-298632863
iteration	
10/2.881초	정상
20/5.859초	정상
30/2.438초	정상
40/2.88초	정상
50/2.163초	-298632863

50을 입력했을 경우 재귀 함수의 경우 엄청나게 오래 시간이 걸렸습니다. 지금까지 배운 것 중에서 모든 소스코드가 오류 없이 컴파일 되고 이상한 값이 뜨고, 음수로 나와있기에 오버플로우?? 말고는 잘 모르겠어서 아마 오버플로우로 인해서 값이 제대로 쓰지 않는게 아닌가 생각이됩니다. 오버플로우가 일어나는 조건이 입력값이 해당하는 값보다 훨씬 큰 값을 입력한 경우에 일어나고 값이 음수인지 양수인지 결정하는 부호비트가 달라졌다고 판단하여 오버플로우일 것 같습니다.

행렬과제 & 도전과제

295	902	315	831
732	628	253	182
544	476	524	482
302	701	321	652

302	544	732	295
701	476	628	902
321	524	253	315
652	482	182	831

831	182	482	652
315	253	524	321
902	628	476	701
295	732	544	302

295	732	544	302
902	628	476	701
315	253	524	321
831	182	482	652

295	902	315	831	302	544	732	295	831	182	482	652	295	732	544	302
732	628	253	182	701	476	628	902	315	253	524	321	902	628	476	701
544	476	524	482	321	524	253	315	902	628	476	701	315	253	524	321
302	701	321	652	652	482	182	831	295	732	544	302	831	182	482	652