1. 다음 주어진 식에 대해 tight한 Big-O estimation을 구하여라. (10점)

a) $5 + 0.1n^3 + 15n$ (2점)

$O(n^3)$

b) $n + 100n^{1.5} + 150n^{1.75}$ (3점)

$O(n^{1.75})$

c) $2log_4 n + 40log_2 log_2 n$ (5점)

$O(\log n)$

2. 다음 주어진 의사코드(psudocode)의 시간 복잡도(time complexity)를 계산하고, 그 근거를 간략히 서술하라. (5점)

```
for (i = n; i > 0; i = i/2) {
    for (j = 1; j < n; j = j * 2) {
        for (k = 0; k < n; k = k + 2) {
            sum += (i + j + k);
        }
    }
}
```

I루프는 $n=2^k$일때 k+1번 수행되므로 $O(\log n)$번

J루프는 $n=2^k$일때 k번 수행되므로 $O(\log n)$번

k루프는 $n=2k$일때 k번 수행되므로 $O(n)$번 수행된다

따라서 sum에 대한 업데이트 기준으로 시간복잡도는 $O((\log n)^2 n)$이다.

4

3. **길이가 각각 1 ~ K 사이인** M **개의** **'이미 정렬되어있는'** **배열**(array)들에 대하여, **merge sort를 이용**하여 **최종적으로 길이가** N **인 하나의 배열**로 합치는 작업을 수행하고자 한다. 이때, 이 작업의 시간 복잡도(time complexity)를 구하고, 그 근거를 간략히 서술하라. (5점)

시간복잡도 : $O(MK \log M)$ 혹은 $O(N \log M)$

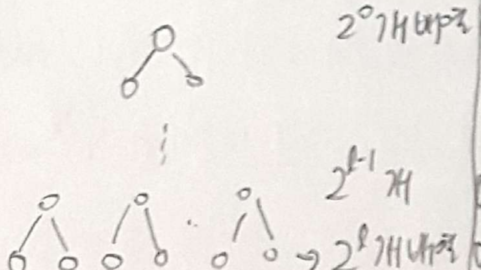
-1

근거 : 간단히 하기위해 $M=2^l$로 잡는다. (아닌경우 더미원소를 채워준다)

$2^l K$개

2K개

(원소) 최대 K개



2^0개 배열 $O(2^l k)$

2^{l-1}개 $O(2^l k)$

2^l개 배열 $O(2^l k)$

merge sort의 merge 연산은 길이 A, B인 배열에 대해서 $O(A+B)$가 되고, 따라서 각 단계에서 드는 비용은 원소개수를 모두 더한것과 같다. 따라서 각단계에서는 $O(2^l k)$가 되고, 이는 l단계만큼 수행되므로 최종시간복잡도는 $O(lk2^l)$이고 m으로나타내면 $O(mk \log m)$이다. 그런데 $km \le N$이고 $k \le \frac{N}{m}$이므로 이는 $O(N \log m)$과 같다

Scanned by CamScanner

4. "*The game of death*" 라는 게임이 있다. 이 게임은 N 명의 사람들이 모여서 진행을 한다. 게임이 시작되면 모든 사람들은 각각 자기 자신을 제외한 $N-1$ 명 중 한 명을 가리킨다. 그리고 N명의 사람 중 한 사람이 2 이상의 정수 K를 말하고, 자기가 가리키고 있던 사람에게 차례를 넘긴다. 지목을 당한 사람은 다시 자기가 가리키고 있던 사람에게 차례를 넘기며 K 번째로 지목을 당한 사람이 나올 때까지 이 과정을 반복하여 벌칙을 수행할 사람을 정하게 된다. 이때, **자신이 이 게임에서 숫자를 부르는 사람**이고, **총 53명의 사람이 이 게임에 참가**한다고 할 때, 자신이 절대 벌칙을 수행하지 않도록 하는 **가장 작은 숫자가 무엇인지** 적고, **그 숫자가 확실히 자신이 지목되지 않도록 함을 증명**하여라. (10점)

$1 \neq K$

가장 작은숫자는 59이다

proof) N명의 사람들이 각자 다른사람을 지목했을 때, N명의 사람은

K명의 사람이 $\underbrace{\circ \to \circ \to \circ \to \cdots \circ}_{K개}$ 이런식으로 순환하며 지목하는 components 와,

이러한 고리에 포함되지 않는 사람들로 나눌 수 있다. 이때 한 component의

최대크기는 N이다. $1<K<58$ 인 자연수를 말했을 때 최악의 경우 각각

K	2	3 …	53	54	55	56	57	58
L	2	3 …	53	2	5	2	3	2

에서 길이가 L에 해당하는 component에 속해있을 때 자신이 걸리게 된다.

이는 길이 L의 component에서 K를 말하면 $T \equiv K \pmod{L}$ 번째에서 넘어간 사람이 걸리게 되기 때문이다. 그런데 59의 경우 $2 \le L \le 53$ 에서 $T \equiv 59 \pmod{L}$ 를 0으로 만드는 정수 L이 없다. (∵소수)

따라서 59를 말하면 걸리지 않는다.

5. 임의의 **양의 정수** n에 대하여, n^3+2n **은 항상 3으로 나누어 떨어짐을 증명**하여라. (10점)

수학적귀납법을 사용한다.

i) basis $n=1$일때, $1+2=3$은 3으로 나누어 떨어진다

ii) inductive step n일때 성립한다 가정하고 ($n \ge 1$) $n+1$ 일때 성립함을 보인다.

가정: $n^3+2n \equiv 0 \pmod 3$ 에서,

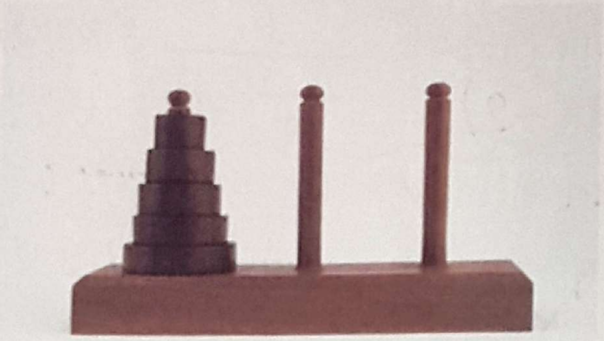
$$(n+1)^3+2(n+1) = n^3+3n^2+3n+1+2n+2$$

$$= n^3+2n+3(n^2+n+1) \text{ 인데,}$$

$n^3+2n+3(n^2+n+1) \equiv n^3+2n \equiv 0 \pmod 3$ 이다. (∵ modular 연산의 성질)

따라서 $(n+1)$에서도 성립하고, 귀납법에 따라 주어진 명제가 성립한다.

6. 다음은 "하노이의 탑" 문제에 대한 설명이다. 아래의 설명을 읽고 질문에 대해 답하여라. (10점)



위의 그림과 같이 **3개의 기둥**이 있고 그 중 하나의 기둥에 *N* 개의 크기가 모두 다른 원반들이 꽂혀있다. 이때, 원반들을 모두 다른 한 기둥으로 옮겨서 꽂는 것이 이 문제의 목표이다. 단, **한 번에 하나의 원반을 움직일 수 있으며, 크기가 더 큰 원반은 더 작은 원반보다 위에 꽂혀있을 수 없다**. 이때, N개의 원반을 한 기둥에서 다른 기둥으로 모두 옮기려면, **최소 몇 번을 옮겨야 하는지** 알아내고자 한다. 마지막 한번

어떻게 풀이

a) 이 문제를 해결하는 재귀적(recursive) 알고리즘을 기술하여라. (5점)

```
int hanoi (int n, int a, int b) //n개를 a원반에서 b원반으로 옮긴다
    if n=1 then {print a+"→"+b; return 1;}
    else { c=6-a-b; //c는 {1,2,3}-{a,b}의 원소
        r=hanoi (n-1, a, c);
        print "\n"+a+"→"+b+"\n"; r++;
        r+=hanoi (n-1, c, b);
        return r;
    }
```

*답은 hanoi(n,1,3)이다
(1→3으로 옮길때)

b) 같은 문제를 반복적(iterative)인 방법으로 해결하는 알고리즘을 기술하여라. (5점)

```
//s[i][j][k]; i개는 j→k로 옮기는 방법 string
// r[i][j][k]; i개는 j→k로 옮기는 operation 수

for i=1 to n
  for j=1 to 3
    for k=1 to 3
      if j==k then continue;
      if i==1 then {s[i][j][k]=j+"→"+k; r[i][j][k]=1;}
      else { l=6-j-k;
          s[i][j][k]=s[i-1][j][l]+"\n"+i+"→"+j+"\n"+s[i-1][l][k]
          r[i][j][k]=r[i-1][j][l]+1+r[i-1][l][k];
      }
```

7. $\frac{m}{n}$이 **무한 소수(infinite decimal)가 되도록** 하는 **임의의 자연수** m, n에 대하여 **그 순환 마디(repetend)의 길이가 항상** n**보다 같거나 작음을 보여라.** (10점)

8

순환마디의 길이가 L인 소수는 $\dfrac{K}{\underbrace{99\cdots9}_{L개}\underbrace{0\cdots0}_{T개}}$ (어떤 정수 K, T, T≥0)로 나타내어진다.

증명없이 이용 −2

따라서 분모 n이 위식의 분모형태로 나타내어짐을(L≤n) 증명하면 된다.

n에대해, $\underbrace{9}_{1개}$, 99, 999, ⋯ $\underbrace{99\cdots9}_{n개}$ 를 생각한다.

i) 이중에서 n으로 나누어떨어지는 꼴이 있다면 그대로 된 것이다 → (나머지 n−1가지 (0제외) 숫자 n개.

ii) 나누어떨어지지 않는다면 비둘기집원리에 의해 A≡B (mod n)인 위의 꼴의 A, B가 하나는 존재한다. WLOG A<B라 가정할 수 있고, B−A는 99⋯90⋯0의 형태를 갖는다 따라서 n은 항상 해당꼴로 나타내어진다.

8. 어떤 사람이 **계단을 오르는데, 한 번에 1칸, 2칸, 또는 3칸을 오를 수 있다**고 한다. n **칸인 계단을 오르는 경우의 수를** A_n 이라고 할 때, 다음 물음에 답하시오. 단, 1칸을 오르고 2칸을 오르는 것과 2칸을 오르고 1칸을 오르는 것은 서로 다른 방법이다. (20점)

a) A_4 의 값을 구하여라. (3점)

$A_4 = 7$ (∵ C의 식)

b) A_8 의 값을 구하여라. (7점)

$A_8 = 81$ (∵ C의 식)

c) A_k 를 점화식(recurrence formula)으로 나타내어라. (10점)

$$A_k = \begin{cases} 1 & k=1 \\ 2 & k=2 \\ 4 & k=3 \\ A_{k-1} + A_{k-2} + A_{k-3} & k \geq 4 \end{cases}$$

k=1에대해 1으로 오르는 한가지

k=2 에대해 1,1 과 2로 오르는 두가지

k=3에대해 1,1,1과 1,2와 2,1와 3으로 오르는 4가지.

k=n에대해, 마지막으로 한칸을 오르면 A_{n-1}, 두칸은 A_{n-2}, 세칸은 A_{n-3} 가지가 되므로 이를 합한다

9. 원형 탁자에 n**개의 자리가 있고** n**명이 각 자리에 한 명씩** 앉아있었다. Dance time이 되어 모두 자리에서 일어났다가 돌아와서 앉았는데, **모두가 다른 사람의 자리에 앉게 되는 경우의 수를** A_n 이라고 할 때, 다음 물음에 답하시오. (20점)

265
4×11 5
265 + 44 = 309
309 × 6 = 1854
1854 + 265 = 2119
2119 × 6 … 14833

2 4 5 6 7 8
0 1 2 9 44 265 1854

a) A_4 의 값을 구하여라. (3점)

$A_4 = 9$이다 (∵ c의 식)

b) A_8 의 값을 구하여라. (7점)

$A_8 = 14833$ (∵ c의 식)

c) A_k 를 점화식(recurrence formula)으로 나타내어라. (10점)

$$A_k = \begin{cases} 0 & k=1 \\ 1 & k=2 \\ (k-1)(A_{k-1}+A_{k-2}) & k>2 \end{cases}$$ 이다.

proof) 강한 수학적귀납법을 사용한다.

i) basis
$k=1$ 에서는 배치방법이 없다. 따라서 0이다
$k=2$ 에서는 A, B에대해 [B, A] 밖에 없다. 따라서 1이다.

ii) inductive
$n \geq 2$ 에서 $n>k$ 의 모든 자연수 k 에대해 식이 성립한다고 가정하고
$A_n = (n-1)(A_{n-1}+A_{n-2})$ 임을 보인다.

case A) $n-1$명의 배치가 끝났을 때, n번째 사람이 각 사람과 서로 자리를 바꿔앉는 경우를 생각한다. $n-1$번 바꿔앉을 수 있고 각각 A_{n-1}개만큼 가짓수가 있다. 따라서 $(n-1)A_{n-1}$개

case B) $n-2$명을 배치하고, x와 n번째 사람이 서로 자리를 바꾸는 경우를 생각한다. case A에서 x에 앉는 x가 n과 자리를 바꾸는 경우는 포함되지 않으므로 이도 더해줘야 한다. $n-1$번 바꿔앉고 각각 A_{n-2}개 만큼 가짓수가 있다. 따라서 $(n-1)A_{n-2}$개.

case A, B에 따라서
$A_n = (n-1)(A_{n-1}+A_{n-2})$
가 성립하고 귀납법에 따라 증명된다