

碩士學位論文

프랙탈 이론을 이용한 게시판 알고리즘  
설계 및 구현

指導教授 李尙俊



濟州大學校 産業大學院

電子電氣工學科

文男元

2003년 6월

프랙탈 이론을 이용한 게시판 알고리즘 설계 및 구현

指導教授 李尙俊

이 論文을 工學 碩士學位 論文으로 提出함

2003年 6月 日

濟州大學校 産業大學院



電子電氣工學科  
컴퓨터工學專攻  
JEJU NATIONAL UNIVERSITY LIBRARY

文男元

文男元の 工學 碩士學位 論文을 認准함

2003年 6月 日

審查委員長 郭 鎬 榮 印

審查委員 金 壯 亨 印

審查委員 邊 翔 庸 印

# 목 차

ABSTRACT .....	1
I . 서론 .....	2
II . 게시판 동작방식에 대한 고찰	
1. 순서갱신형 게시판 .....	3
2. 고정위치형 게시판 .....	7
3. 연결리스트 형태의 게시판 .....	11
4. 게시판간 장단점 분석 .....	15
III . 프랙탈 이론을 이용한 게시판 구현	
1. 프랙탈(Fractal) .....	16
2. 게시판 알고리즘을 위한 프랙탈 .....	21
3. 프랙탈 이론을 이용한 게시판 구현 .....	24
IV . 결과	
1. 테스트 전제조건 .....	28
2. 입력속도 비교 .....	29
3. 출력속도 비교 .....	32
V . 결론 및 향후 연구 .....	33
참고문헌 .....	34

## [표 차례]

Table. 1. 순서갱신형 게시판의 DB구조 .....	4
Table. 2. 고정 위치형 게시판의 DB 구조 .....	8
Table. 3. 연결리스트 형태 게시판의 DB 구조 .....	12
Table. 4. 프랙탈 이론을 사용하는 게시판의 DB구조 .....	24

## [그림 차례]

Fig. 1. 순서갱신형 게시판의 트리구조 .....	3
Fig. 2. 순서갱신형 게시판의 출력 모양 .....	3
Fig. 3. 순서갱신형 게시판의 답글 입력 순서도 .....	5
Fig. 4. 순서갱신형 게시판의 게시물 출력 순서도 .....	6
Fig. 5. 고정 위치형 게시판의 트리구조 .....	7
Fig. 6. 고정위치형 게시판의 출력 모양 .....	7
Fig. 7. 고정위치형 게시판의 답글 입력 순서도 .....	9
Fig. 8. 고정위치형 게시판의 게시물 출력 순서도 .....	10
Fig. 9. 연결리스트 형태 게시판의 트리구조 .....	11
Fig. 10. 연결리스트 형태 게시판의 출력 모양 .....	11
Fig. 11. 연결리스트 형태 게시판의 답글 입력 순서도 .....	13
Fig. 12. 연결리스트 형태 게시판의 게시물 출력 순서도 .....	14
Fig. 13. 코흐눈송이 .....	19
Fig. 14. 시어핀스키 삼각형 .....	19
Fig. 15. 페아노 곡선 .....	19
Fig. 16. 줄리아 집합 .....	20
Fig. 17. 만델브로트 집합 .....	20
Fig. 18. 트리구조를 갖는 기초 프랙탈 .....	21
Fig. 19. 프랙탈 모형 .....	22
Fig. 20. 프랙탈 알고리즘을 이용한 게시판의 답글 입력순서도 .....	25
Fig. 21. 프랙탈 알고리즘을 이용한 게시판의 게시물 출력 순서도 .....	26
Fig. 22. 프랙탈 알고리즘을 이용한 게시판 출력모양 .....	27
Fig. 23. 게시판간 입력속도 비교 그래프 .....	29
Fig. 24. 게시판간 답글 수평달기 시간 비교 그래프 .....	30
Fig. 25. 게시판간 답글 수직달기 시간 비교 그래프 .....	31
Fig. 26. 게시판간 리스트 출력시간 비교 그래프 .....	32

**The Design and Implementation of Bulletin Board algorithm  
based on fractal theory**

**Nam-Weon Mun**

*Department of Electrical and Electronic Engineering Graduate School  
of Industry*

*Cheju National University*

*Supervised by Professor Sang-Joon Lee*



**ABSTRACT**

We analyze bulletin board application currently used, and we propose the method that minimize disk I/O to enhance the I/O efficiency of the transmission of a large size of data. To improve the performance, we use fractal algorithm. Experiential result shows better performance than the other algorithm.

Our next researches include the effort to improve the speed of computation by converting floating point operation to integer operation. And the method to enhance the precision of calculation is included also.

## I. 서론

현재 인터넷상에서 게시판의 용도는 의사전달 및 지식 공유의 장으로써 대단히 폭넓게 활용되고 있다. 많은 커뮤니티 사이트가 수십만 개의 게시판을 운영하고 있으며 '네이버 지식in'인 경우 하루 2만 건 이상의 입력과 수십만 번의 조회가 이루어지고 있으며 이를 통한 정보공유가 활발히 진행되고 있다.

하지만 게시판의 데이터가 축적됨에 따라 게시판이 대용량화 되어 가고 입력과 출력의 속도가 저하되고 있으며 알고리즘 개선보다는 시스템의 분산과 고성능화를 꾀하여 서비스의 질을 유지하고 있다.

이에 좀더 입출력 속도를 향상시킬 수 있는 게시판 개선 알고리즘 연구를 통하여 저비용으로 서비스를 유지할 수 있도록 한다.

일반적인 게시판의 아래와 같은 조건을 만족해야 한다.

첫째, 답글의 개수와 깊이를 충분히 입력할 수 있어야 한다.

둘째, 데이터 액세스를 최소한으로 줄여야 한다.

셋째, 수정 및 삭제가 용이해야 한다.

위의 조건을 만족하며 좀더 효율적인 알고리즘을 가진 게시판을 구현하고자 기존의 게시판의 입,출력 알고리즘을 분석하고 이를 응용 발전시키는 방향으로 알고리즘을 만들어 보고자 한다.

## II. 게시판 동작방식에 대한 고찰

### 1. 순서갱신형 게시판 (Board01)

#### 1) 개요

- 입력시 인덱스된 필드를 기준으로 하여 자신의 위치를 파악하고 위치 이후의 인덱스 번호를 일괄 갱신하여 새로운 인덱스를 생성한다. 출력시는 인덱스번호를 기준으로 하여 게시물을 나열한다.

#### 2) DB 및 트리 구조

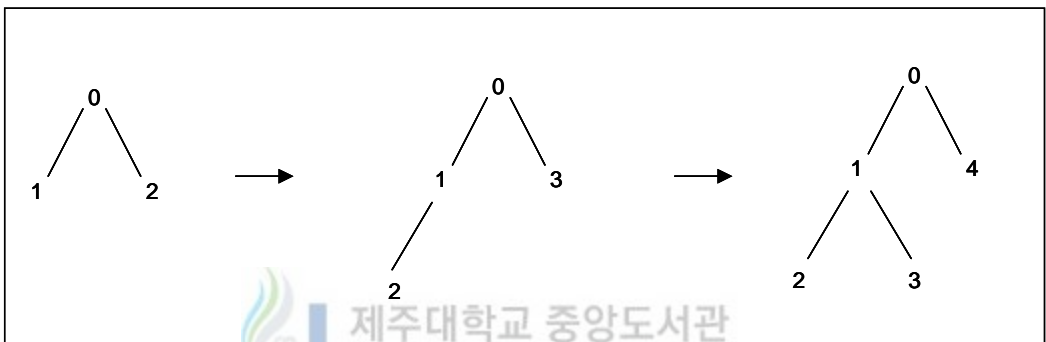


Fig. 1. 순서갱신형 게시板的 트리구조

**정렬갱신형 게시판(board01)**

기본게시판 | 정렬 | 수직일기 | 답글 | 수평일기 | 답글 | 수직일기

기본게시판

no	uid	idx	depth	seqno	subject	name	counter	indate	p_time
2	2	0	0	0	두번째 원글	test	2	03/05/22	0.0031229257583618
3	2	1	1	0	Re:1 두번째 원글->1	test	2	03/05/22	0.012629805889326
5	2	2	2	0	Re:2 두번째 원글->1->1	test	0	03/05/22	0.009906888081177
6	2	3	2	0	Re:2 두번째 원글->1->2	test	1	03/05/22	0.0099489688873291
7	2	4	3	0	Re:3 두번째 원글->1->2->1	test	0	03/05/22	0.010446906889783
4	2	5	1	0	Re:1 두번째 원글->2	test	0	03/05/22	0.012389063835144
1	1	0	0	0	test	test	1	03/05/19	0.0025449951226196

<< || / 총 1페이지 >>

리스트 출력시간 : 0.044952018358323 초

Fig. 2. 순서갱신형 게시板的 출력 모양

Field	Type	Key	Extra
no	int(11)	PRI	auto_increment
uid	int(11)		원글순서번호
idx	int(11)		원글정렬번호
depth	int(11)		답글의 깊이
seqno	int(11)		깊이에 따른 순서
indate	int(11)		입력일자
name	varchar(20)		글쓴이
passwd	varchar(20)		비밀번호
email	varchar(50)		이메일
subject	varchar(100)		글의 제목
text	text		글의 내용
counter	int(11)		조회수
p_time	double		입력소요시간

Table. 1. 순서갱신형 게시판의 DB구조

### 3) 입출력 알고리즘

#### 3-1) 원글의 입력

- ① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.
- ② 최종번호에 하나를 더한후 정렬순서번호와 글의깊이를 0으로 하는 레코드를 삽입한다.

#### 3-2) 답글의 입력



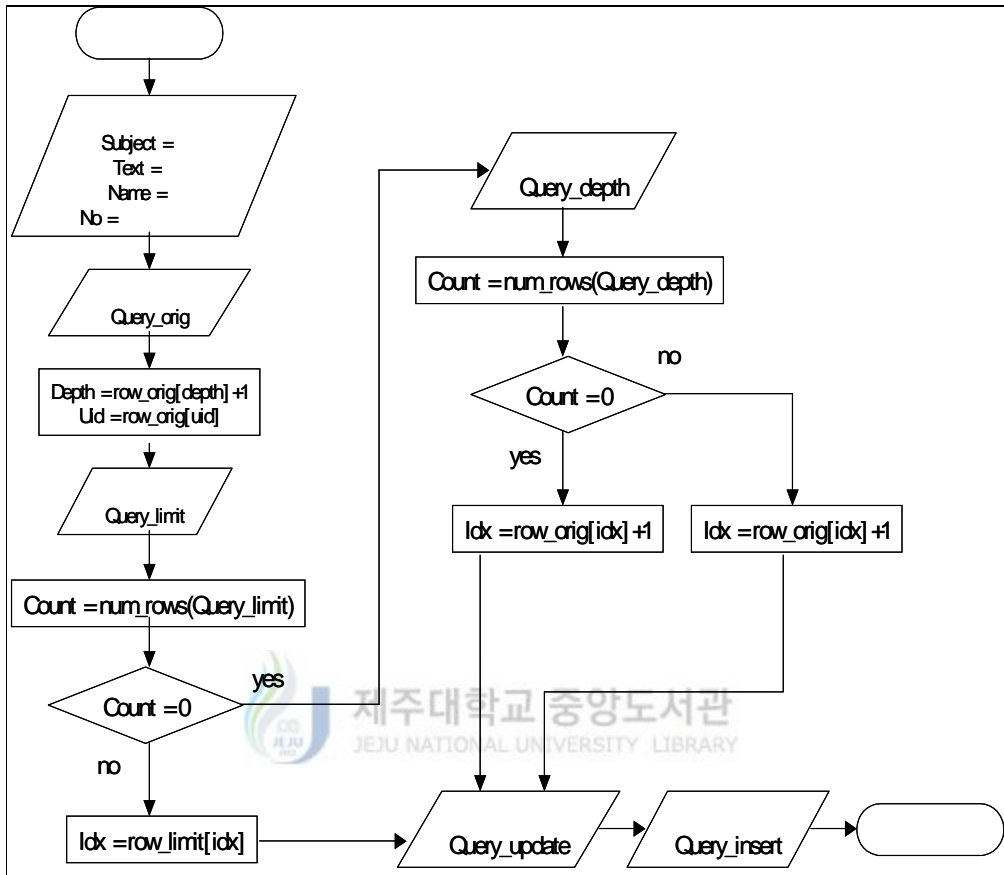


Fig. 3. 순서갱신형 게시판의 답글 입력 순서도

- ① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.
- ② 본글의 깊이와 본글순서를 이용하여 답글의 깊이와 본글순서를 정한다.
- ③ 정렬순서번호가 본글보다 크고 본글과 같은 깊이의 기존글이 있는지 검색하고 있을 경우 기존글의 정렬순서번호를 가져온다.
- ④ 본글과 같은 깊이의 기존글이 없을 경우 답글의 깊이보다 깊고 본글의 정렬 순서번호보다 가장큰 정렬순서번호를 가지는 기존글의 정렬순서번호를 가져온다.
- ⑤ 위의 경우가 없을 경우 본글의 정렬순서번호에 하나를 더하여 답글의 정렬순서번호를 정한다.
- ⑥ 답글의 정렬순서번호보다 같거나 큰 정렬순서번호를 가지는 모든글에 대해 정렬순서번호를 하나씩 증가시킨다.
- ⑦ 답글을 삽입한다.

### 3-3) 게시물의 출력

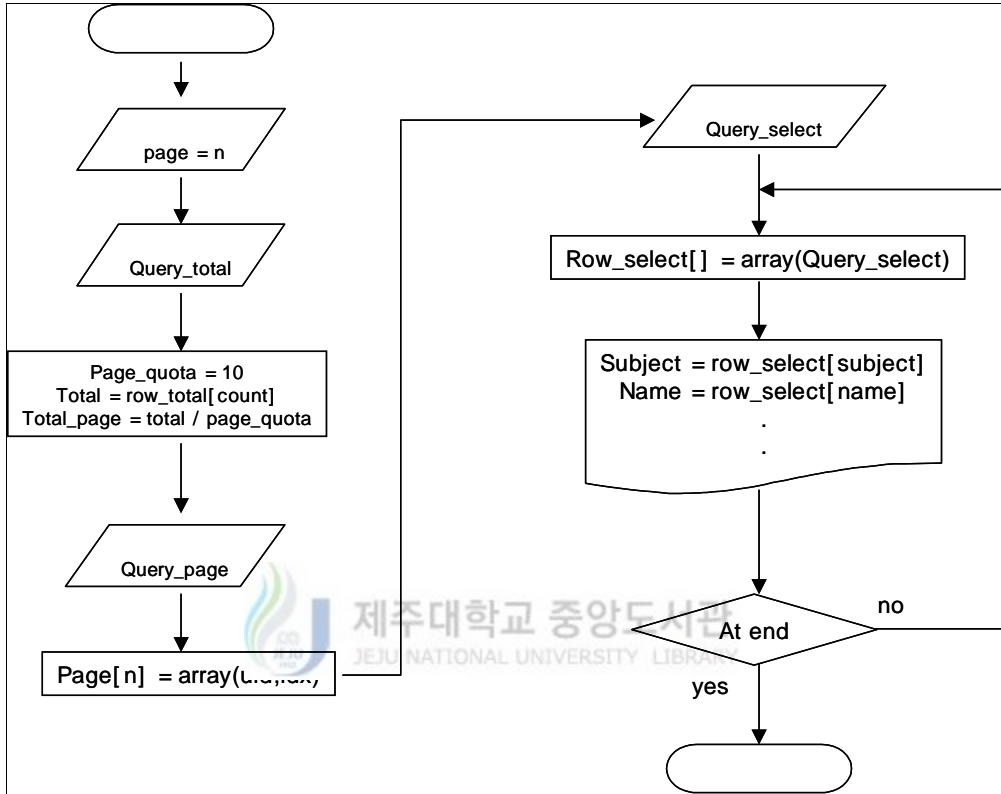


Fig. 4. 순성갱신형 게시판의 게시물 출력 순서도

- ① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지 당 글의 개수로 나누고 전체 페이지를 산출한다.
- ② 페이지당 작업을 위하여 원글번호의 역순과 정렬번호순으로 원글번호와 정렬 순서번호를 가져온다.
- ③ 각페이지당 시작글의 원글번호와 정렬순서번호를 배열에 배당한다.
- ④ 검색하고자 하는 페이지의 시작글의 원글번호와 정렬순서번호를 배열에서 찾고 페이지당 글의개수 만큼 테이블에서 불러내어 출력한다.
- ⑤ 출력 질의

Query="select \* from \$table where (uid=\$a and idx>='\$b') or (uid <\$a) order by uid desc,idx limit \$page\_quota" ;

## 2. 고정위치형 게시판 (Board02)

### 1) 개요

- 입력시 기본글과 답글사이에 관계번호와 답글의 위치번호를 넣고 관계번호를 우선하여 문자로된 위치번호순으로 글을 가져온다.

### 2) DB 및 트리 구조

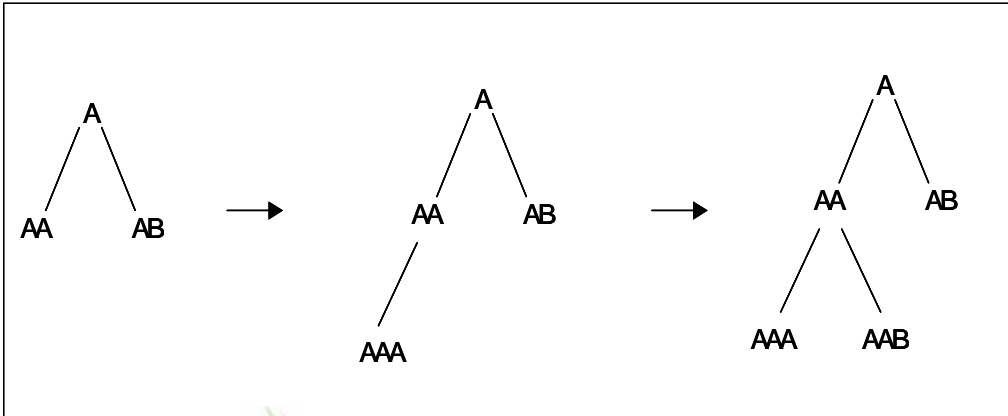


Fig. 5. 고정 위치형 게시판의 트리구조

고정위치형 게시판(board02)

기본게시판 | 한글 추적알기 | 답글 수명알기 | 답글 추적알기

기본게시판

no	uid	thread	depth	subject	name	counter	lndate	p_time
2	2	A	0	두번째 원글	test	3	03/05/22	0.0028879642486572
3	2	AA	1	▶ 두번째 원글->1	test	2	03/05/22	0.0073080062966211
5	2	AAA	2	▶▶ 두번째 원글->1->1	test	0	03/05/22	0.0071438743041992
6	2	AAB	2	▶▶▶ 두번째 원글->1->2	test	1	03/05/22	0.0077109306853027
7	2	AABA	3	▶▶▶▶ 두번째 원글->1->2->1	test	0	03/05/22	0.0072219371795654
4	2	AB	1	▶ 두번째 원글->2	test	0	03/05/22	0.007999062538147
1	1	A	0	원글의 제목	test	0	03/05/19	0.002932071685791

◀ ||| / 총 1페이지 ▶

리스트 출력시간 : 0.043442010879517 초

글쓰기

Fig. 6. 고정위치형 게시판의 출력 모양

Field	Type	Key	Extra
no	int(11)	PRI	auto_increment
uid	int(11)		원글순서번호
thread	varchar(8)		문자위치 번호순서
depth	int(11)		글의 깊이
indate	int(11)		입력일자
name	varchar(20)		글쓴이
passwd	varchar(20)		비밀번호
email	varchar(50)		이메일
subject	varchar(100)		글의 제목
text	text		글의 내용
counter	int(11)		조회수
p_time	double		입력소요시간

Table. 2. 고정 위치형 게시판의 DB 구조

### 3) 입출력 알고리즘

#### 3-1) 원글의 입력

- ① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.
- ② 최종번호에 하나를 더한후 문자위치번호순서를 "A"로 하고 글의깊이를 0으로 하는 레코드를 삽입한다.

#### 3-2) 답글의 입력

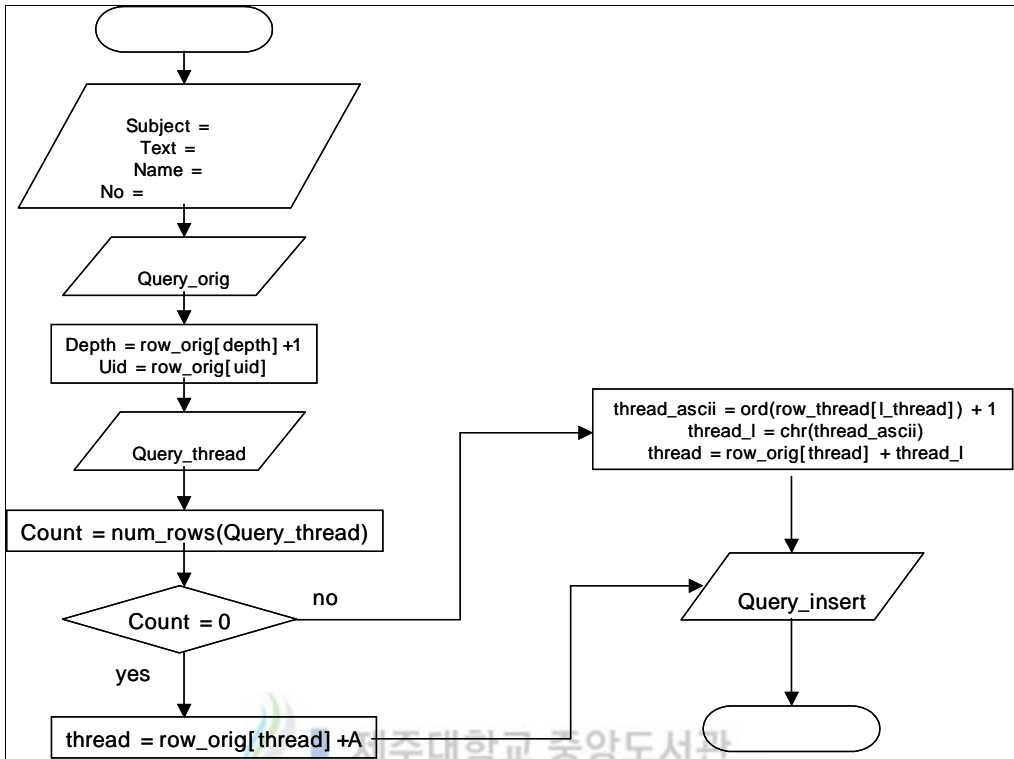


Fig. 7. 고정위치형 게시판의 답글 입력 순서도

- ① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.
- ② 본글의 깊이를 이용하여 답글의 깊이를 정의한다. 답글의 깊이가 7을 넘을 경우 에러를 표시하고 작업을 빠져나간다.
- ③ 답글의 깊이와 깊이가 같고 본글의 문자위치번호순서와 같은 순서를 갖는 마지막 기존글을 가져온다.
- ④ 위의 마지막 기존글이 없을 경우 본글의 문자위치번호에 "A"를 더한다.
- ⑤ 마지막 기존글의 마지막 문자열을 아스키값으로 변환하여 이에 1을 더하고 이값이 90을 넘을 경우 에러를 표시하고 작업을 빠져나간다.
- ⑥ 아스키값을 문자로 변환하고 본글의 문자위치번호에 문자를 더한다.
- ⑦ 답글을 삽입한다.

### 3-3) 게시물의 출력

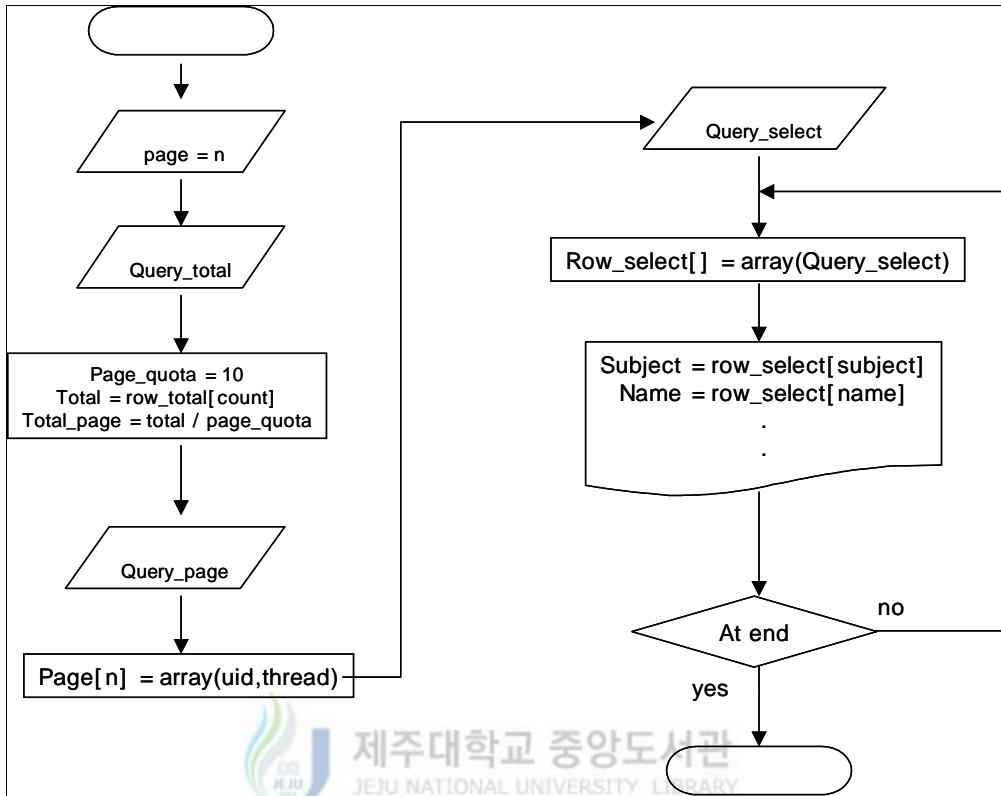


Fig. 8. 고정위치형 게시판의 게시물 출력 순서도

- ① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지 당 글의 개수로 나누고 전체 페이지를 산출한다.
- ② 페이지당 작업을 위하여 원글번호의 역순과 문자위치번호순으로 원글번호와 문자위치번호를 가져온다.
- ③ 각페이지당 시작글의 원글번호와 문자위치번호를 배열에 배당한다.
- ④ 검색하고자 하는 페이지의 시작글의 원글번호와 문자위치번호를 배열에서 찾고 페이지당 글의개수 만큼 테이블에서 불러내어 출력한다.
- ⑤ 문자위치번호의 우선순위는 A > AA > AAA > AB > ABA 가 된다.
- ⑥ 출력 질의

Query="select \* from \$table where (uid=\$a and thread>='\$b') or (uid <\$a) order by uid desc,thread limit \$page\_quota" ;

### 3. 연결리스트 형태의 게시판 (Board03)

#### 1) 개요

- 입력시 본문과 답글사이에 순회 할 수 있는 자식노드를 넣고 출력시에 노드를 따라서 글을 가져온다

#### 2) DB 및 트리 구조

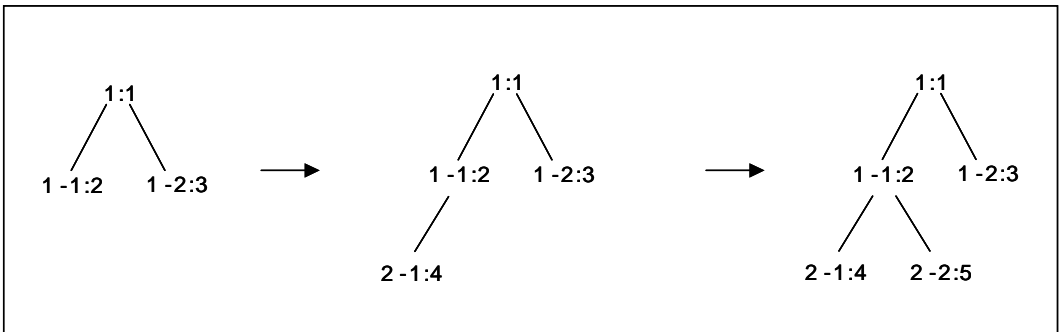


Fig. 9. 연결리스트 형태 게시판의 트리구조

제주대학교 중앙도서관  
JEJU NATIONAL UNIVERSITY LIBRARY

기본게시판
원글 수직달기
답글 수행달기
답글 수직달기

기본게시판

no	uid	cnode	pid	seqno	depth	subject	name	counter	indate	p_time
2	2	Y	0	0	0	두번째 원글	test	3	03/05/22	0.0029379642496572
3	0	Y	2	1	1	Re*1 두번째 원글->1	test	2	03/05/22	0.0099589624676514
5	0	N	3	1	2	Re*2 두번째 원글->1->1	test	0	03/05/22	0.0098530054032407
7	0	N	3	2	2	Re*2 두번째 원글->1->2	test	0	03/05/22	0.0093740224838257
4	0	Y	2	2	1	Re*1 두번째 원글->2	test	1	03/05/22	0.0090399860545044
6	0	N	4	1	2	Re*2 두번째 원글->2->1	test	0	03/05/22	0.0098448991775513
1	1	N	0	0	0	원글의 제목	test	0	03/05/19	0.0024299621582031

◀ 11 / 총1페이지 ▶


이동

리스트 출력시간 : 0.058403015136719 초

글쓰기

Fig. 10. 연결리스트 형태 게시판의 출력 모양

Field	Type	Key	Extra
no	int(11)	PRI	auto_increment
uid	int(11)		원글순서번호
cnode	char(1)		자식노드 유무
pid	int(11)		부모노드의 번호(no)
seqno	int(11)		깊이에 따른 순서
depth	int(11)		글의 깊이
indate	int(11)		입력일자
name	varchar(20)		글쓴이
passwd	varchar(20)		비밀번호
email	varchar(50)		이메일
subject	varchar(100)		글의 제목
text	text		글의 내용
counter	int(11)		조회수
p_time	double		입력소요시간

Table. 3. 연결리스트 형태 게시판의 DB 구조

### 3) 입출력 알고리즘

#### 3-1) 원글의 입력

- ① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.
- ② 최종번호에 하나를 더한후 모노드,깊이,깊이에 따른 순서를 "0"으로 하고 자식노드를 "N"으로 하는 레코드를 삽입한다.

#### 3-2) 답글의 입력



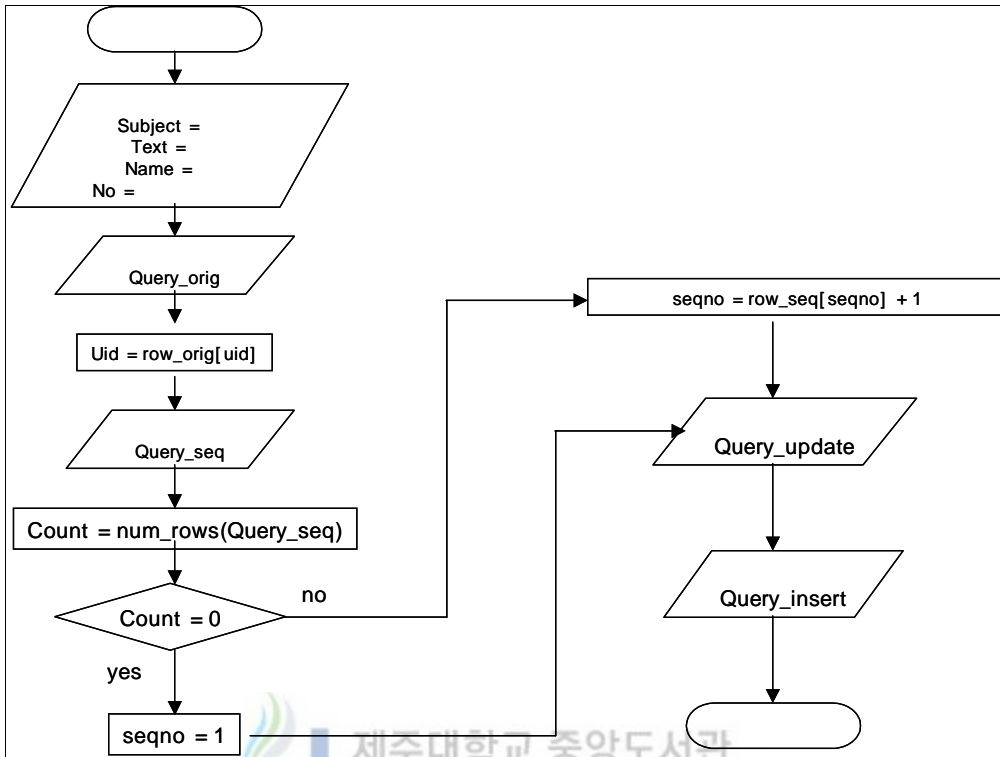


Fig. 11. 연결리스트 형태 게시판의 답글 입력 순서도

- ① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.
- ② 본글의 절대번호를 부모노드로 하는 글들의 깊이에 따른 순서(seqno)중 기존글의 제일 큰번호를 가지고 온다.
- ③ 위의 번호가 없을 경우 순서번호를 1로 하고 있을 경우 기존글의 번호에 1을 더한 순서번호를 갖는다.
- ④ 본글의 깊이에 하나를 더한 깊이를 갖고 본글의 자식노드 유무를 “Y”로 변경한다.
- ⑤ 답글을 삽입한다.

### 3-3) 게시물의 출력

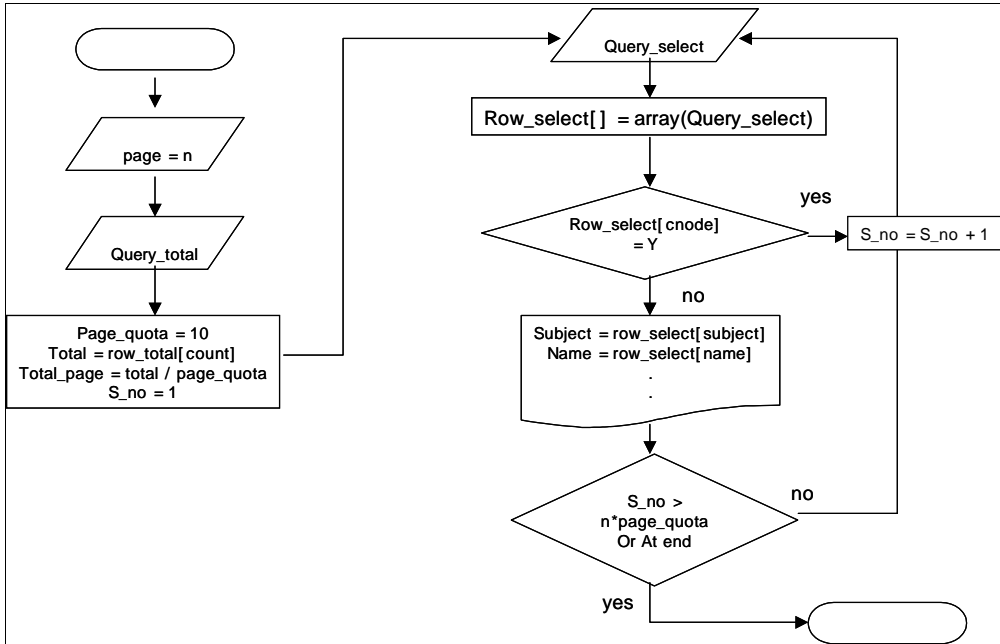


Fig. 12. 연결리스트 형태 게시판의 게시물 출력 순서도

- ① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지 당 글의 개수로 나누고 전체 페이지를 산출한다.
- ② 먼저 원글만을 원글번호의 역순으로 가져온다.
- ③ 원글에 자식노드가 있을 경우 자식노드를 가져오는 함수 get\_cnode()를 실행한다.
- ④ get\_cnode는 자식노드가 없을때까지 실행되는 재귀함수이다.
- ⑤ 출력번호(s\_no)가 원하는 페이지의 출력번호보다 높을 경우 출력을 중지한다.
- ⑥ 출력 질의

Query\_no="select no,cnode from \$table where uid != 0 order by uid desc" ;

#### 4. 게시판간 장단점 분석

- 정렬갱신형 게시판은 가장 빠른 속도로 출력 할 수 있으나 입력시 최악의 경우 전체글의 정렬번호를 갱신하여야 하며 이때 입력 속도의 저하가 발생한다.
- 고정위치형 게시판은 글의 개수와 무관하게 입출력 속도를 보장할 수 있으나 답글의 개수와 깊이가 제한된다.
- 연결리스트형 게시판은 무한 확장 할 수 있으며 안정된 입력속도를 보장하지만 입력된 글의 개수와 비례하여 출력의 속도가 현저하게 저하된다.



### III. 프랙탈 이론을 이용한 게시판 구현

#### 1. 프랙탈(Fractal)

만델브로트(Mandelbrot, 1975)는 해안선, 나무의 모습, 강의 모양을 일반화하는 목적으로 프랙탈이라는 개념을 발표했다. 유클리드기하의 곡선은 확대하면 직선의 모양이 되고, 구는 평면에 가까워지는데, 프랙탈의 해안선, 산악, 구름들을 살펴보면, 표면에서의 울퉁불퉁한 것이 세부구조로 들어가면서 반복적으로 나타난다.

프랙탈의 개념은 조각나거나 가지친 자연구조의 배열뿐 아니라 브라운 운동(Brown motion)에서부터 물방울 운동에 이르기까지 구조의 역동적인 성질들을 묘사하는데 사용될 수 있다. 프랙탈은 과학자들이 자연현상을 측정하는데 사용할 수 있는데, 예를 들면 전기를 전도하는 방식들을 연구하는 데 사용할 수 있다. 그러나 수학 프랙탈은 자연적인 물체에서는 실제로 발견할 수 없는 성질들을 가지고 있다. 무한히 되풀이해서 확대되면서 똑같이 보이는 구조는 실제로 없다. 그럼에도 불구하고 프랙탈 모델은 적어도 한정된 범위에서 실제와 비슷한 접근 방법을 제공해 준다.



프랙탈은 Yourk 와 Lee의 카오스 이론이 내포된다. 카오스 계의 운동이 로렌즈(Lorenz)끌개의 한쪽 날개 위를 돌다가 불규칙적으로 다른 쪽 날개로 넘어가는 것을 반복하는 형태로 나타나는데, 똑같은 길을 반복하지 않으면서 일정한 형태(나비 날개 모양)를 유지한다. 이 기이한 끌개(strange attractor)는 무한히 많은 층으로 이루어졌으나 자체유사성을 지닌, 매우 기묘한 기하학적구조를 가지고 있으며 이 위에서 카오스 운동이 일어나게 된다.

이러한 점에서 동적 운동으로서의 카오스를 이해하기 위하여 공간적 기하학적 구조를 가지고 있으며 카오스 운동이 일어나게 된다. 이러한 점에서 동적 운동으로서의 카오스를 이해하기 위하여 공간적 기하학적 프랙탈 연구가 필요하다. 끌개의 프랙탈 구조와 카오스 운동은 마치 밀가루 반죽과 같이 상태공간을 늘리고 접는 과정의 무한한 반복에 의해 만들어진다. 끊임없는 팽창과 접힘 과정에 의해 초기의 미세한 차이가 크게 증폭되는 현상을 초기조건에 대한 민감한 의존성이라고 한다. 즉 북경에서 나비 한 마리의 날개짓의 팔랑거림이 계속 증폭되어 오랜 시간이 지난 후 지구 반대쪽 뉴욕에서 폭풍우를 불러올 수 있다는 것이다. 이를 나비효과라 한다.

나비효과는 기상모델에서의 비예측성을 낳게 되며 장기예보가 근본적으로 불가능하다는 것을 가르쳐 준다. 컴퓨터를 통하여 로렌쯔의 기상 모델뿐 아니라 다양한 자연현상에서 여러 가지 기묘한 기하학적 구조의 기이한 끝개들이 발견되었다. 즉, 프랙탈 이론은 카오스 속의 변화하지 않는 부분만 착안하여 법칙으로 끌어낸 것이다. 프랙탈은 미분이 가능하여 정규적 모양을 지닌 유클리드 기하와는 달리, 비정규적 구조를 가진다. 이러한 구조를 확대하면 전체의 모습과 비슷한 구조가 다시 나타난다. 프랙탈이란 밖으로 열린 순서적으로 된 패턴을 나타내는 도형으로, 확대해 가면 반복적으로 매우 닮은 세부적 구조가 보이는 특징을 가진다.

1970년대에 요크와 이천암의 수학논문에서 처음으로 '카오스'라는 단어가 사용되었다. 1975년 만델브로트(Mandelbrot)가 '프랙탈한 대상 모양, 우연, 차원'이라는 책을 출판하였다. 이것은 각각 수학과 과학 세계에 충격을 주었다.

프랙탈은 70년대 말부터 물리학자, 지리학자, 건축, 미술, 철학 등의 분야에서 주목을 받게 되었다. 프랙탈에 대한 관심을 갖게 된 것은 컴퓨터의 발달과 더불어 프랙탈 도형을 많은 사람들이 즐길 수 있게 된 것과, 물리학과 관측기술의 진보가 자연 속에 있는 프랙탈한 모양을 만드는데 성공한 데 원인이 있다. 만델브로트는 처음으로 프랙탈에 대해 많은 연구를 시작한 사람으로 자신이 생각한 형상, 차원 및 기하학에 대한 이름을 생각하고, 라틴어의 '부서지다'라는 뜻의 동사 'frangere'에서 파생한 형용사 'fractus'를 찾았다. fractus란 '온전한 것이 아닌', '어중간한', 뜻으로 어원이 같은 영어 단어 'fracture'와 'fraction'의 어감도 적절한 것으로 생각했다. 만델브로트는 영어이면서 불어이며, 명사이자 형용사인 'fractal'을 만들었다.

80년대에 프랙탈이론은 심미적인 기하학 이론으로 연구되어 왔을 뿐 아니라 매개변수의 변화에 정의되는 동력계(dynamics)연구에 있어서 카오스의 본질을 파악하는데 기초를 이루게 된다. 순수수학에서의 단순한 개념으로 출발한 프랙탈은 현대과학의 많은 문제점 기술에 있어서 중요한 위치를 차지하게 되었고, 카오스 문제를 이해하는 데 새로운 혁신적인 시각을 제공하게 된다.

카오스 연구에 빛을 비추기 시작한 포앙카레는 18세기말부터 19세기초에 걸쳐 활약한 프랑스 수학자이다. 그는 뉴턴의 세계관이라고 할 수 있는 태양계에서 카오스를 발견한다. 그러나 그는 결정론적 세계관을 가졌고, 실제의 천문현상은 뉴

턴의 법칙을 잘 만족시키고 있었기에 포앙카레는 결과에 대해 이상하게 생각했다.

그 후에 로렌쯔는 2차 대전 중 기상예보관으로 일하면서 기후에는 어떤 법칙이 존재할 것이라 생각했다. 하루 이틀 후의 일기예보조차 불확실하고 1주일 후의 상황은 전혀 예측할 수 없는 이유가 나비효과 때문인 것을 알았다. 1963년 대류에 관한 방정식을 분석해서 중요한 요소만 남겨 단순한 형태로 만들었는데 비선형요소를 포함하고 있었다. 로렌쯔는 방정식을 풀어가던 중 그 속에 포함되어 있는 정교한 기하학적 구조를 발견하였다. 똑같은 자리로 되돌아오지는 않지만 거의 비슷하게 반복되는 로렌쯔 끌개의 모습이였다. 그러나 포앙카레처럼 주변 과학자들의 관심을 끌지는 못했다.

1960 년대에 몇몇 과학자들은 카오스를 연구하는데 기초가 될 만한 것을 마련했는데, 스테판 스메일(Stephen Smale)이 대표적이다. 그는 동력학계에 위상수학을 결합시키는 연구를 했다.

요크는 로렌쯔가 주장한 '초기 조건의 민감성'은 일상생활의 도처에 존재한다고 생각했다. 요크는 생물학자인 로버트 메이(Robert May)와의 공동연구에서 카오스계에서 나타난 질서를 찾았다.

만델브로트는 자연의 경향성을 밝히려 했고, 사회의 복잡한 무질서 속에서 일정한 질서가 있음을 찾으려 했다. 이와 같은 질서는 뉴턴역학에서 보여지는 단순 명쾌한 질서는 아니었다. 그의 업적은 자연이 가지고 있는 자체 유사성에 대한 연구에서 절정에 이르게 된다. 카오스는 현재 비선형 동력학 이론과 실험도구로서의 컴퓨터 발전과 맞물려 성장하고 있다. 또한 카오스는 수학, 물리학, 생물학, 화학, 지질학, 공학, 생태학, 사회학, 경제학, 과학 철학 등 과학 및 사회 전반에 걸쳐 근본적인 사고의 변화를 가져오고 있으며, 현재공학, 산업에서의 응용이 매우 활발하다.

언제나 부분이 전체를 닮는 자기 유사성(self-similarity)과 소수(小數)차원을 특징으로 갖는 형상. '프랙탈' 이란 이름은 1975년 B.B.만델브로트에 의해 지어졌으나, 이러한 형상들에 관한 추상적 논의는 훨씬 이전부터 있었다. 칸토르집합, 코흐눈송이(Fig.13), 시어핀스키삼각형(Fig.14), 페아노곡선(Fig.15), 줄리아 집합(Fig.16) 등이 그 예이다. 만델브로트가 정의한 만델브로트집합(Fig.17)은 현대에 소개된 프랙탈의 예이다.

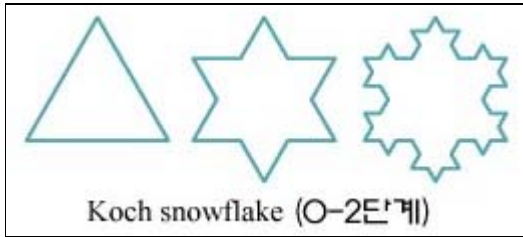


Fig. 13. 코흐눈송이

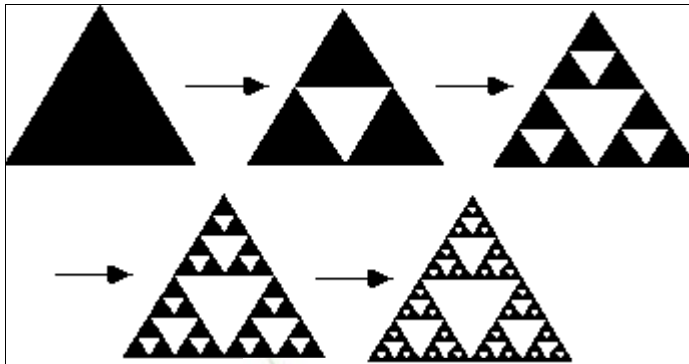


Fig. 14. 시어핀스키 삼각형

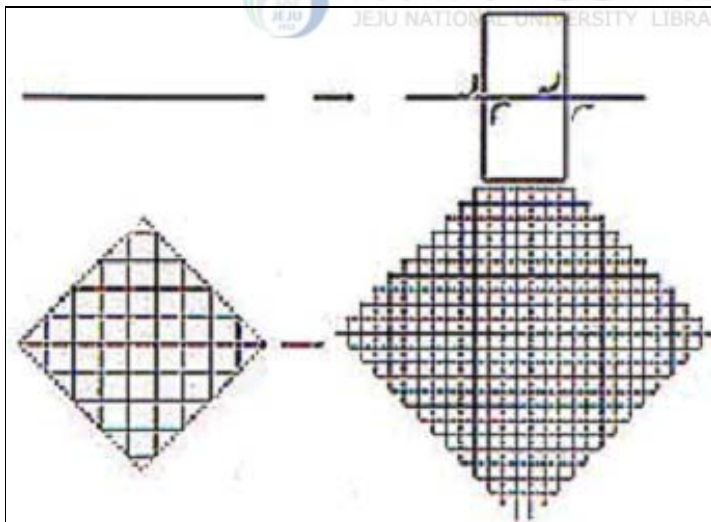


Fig. 15. 페아노 곡선

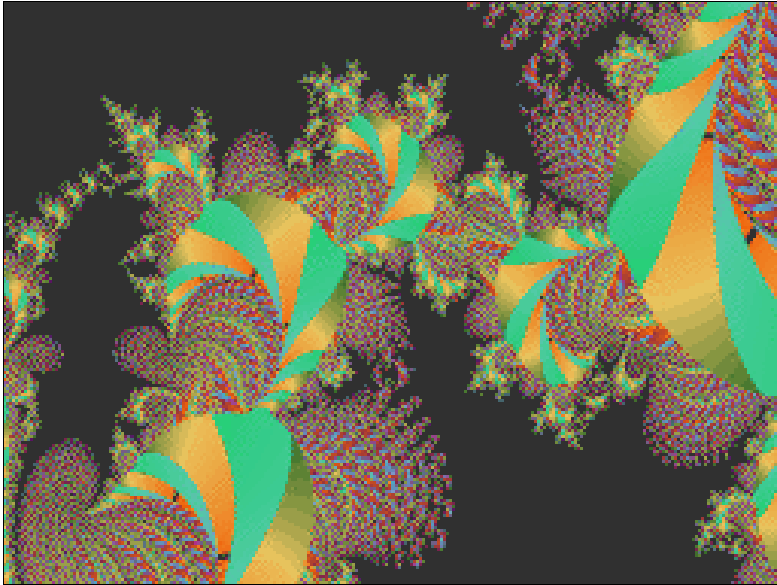


Fig. 16. 줄리아 집합

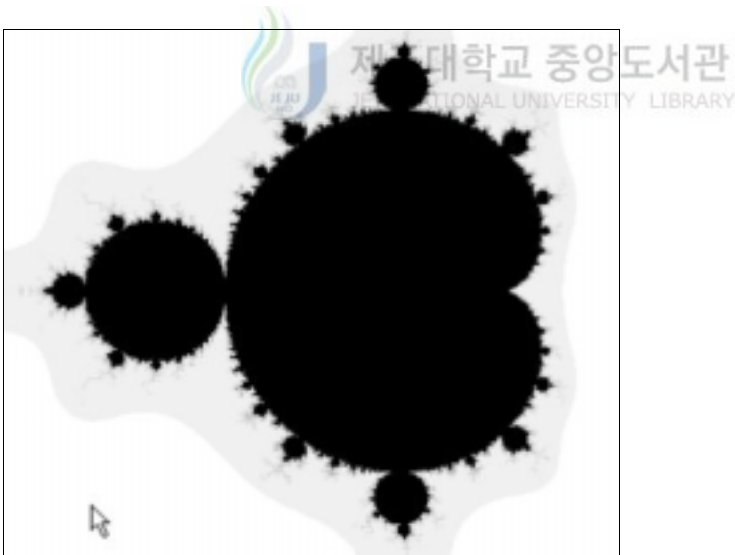


Fig. 17. 만델브로트 집합



## 2. 게시판 입출력을 위한 프랙탈

### 1) 기초 프랙탈



Fig. 18. 트리구조를 갖는 기초 프랙탈

2) 계시관 알고리즘을 위한 프랙탈

① 기본모양

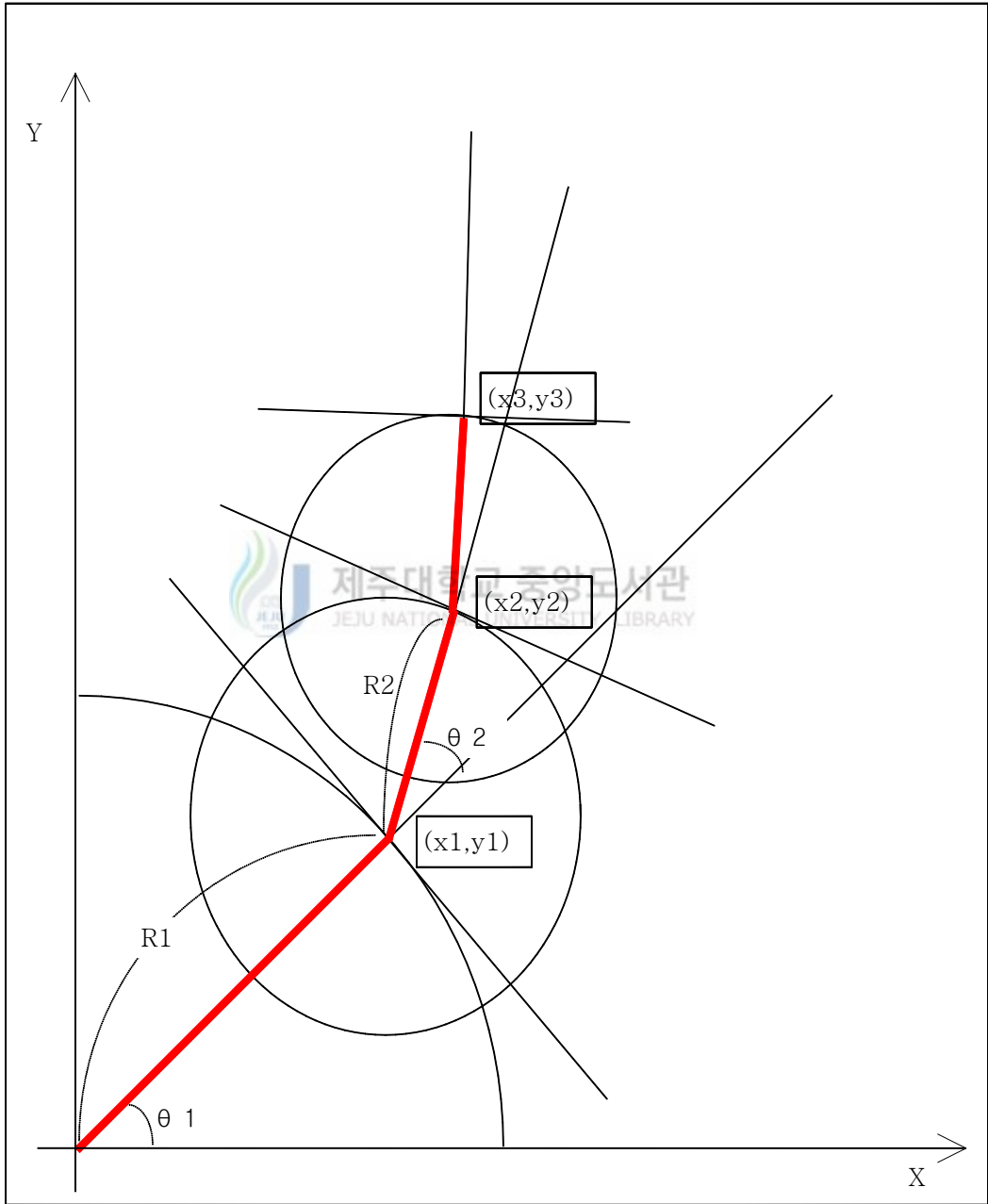


Fig. 19. 프랙탈 모형

② 함수의 정의

- 맨 처음 가지는 길이“R1”과 각“θ1”를 가지는 선분을 그린다.

$$x_1 = R_1 * \cos(\theta_1)$$

$$y_1 = R_1 * \sin(\theta_1)$$

- 두 번째의 가지는 길이와 각이 일정하게 줄어드는 선분을 그린다. 두 번째 가지의 길이(R2)는  $R_1 * (\text{VAR\_RED\_R})$ 가 되고 각“θ2”는  $\theta_1 * (\text{VAR\_RED\_T})$ 가 된다. 여기서 VAR\_RED\_R 은 선분이 줄어드는 비율이고 VAR\_RED\_T는 길이에 따라 각이 줄어드는 비율이다.

$$x_2 = x_1 + R_2 * \cos(\theta_1 + \theta_2)$$

$$y_2 = y_1 + R_2 * \sin(\theta_1 + \theta_2)$$

- 세 번째의 가지는 길이와 각이 동일비율로 줄어드는 선분을 그린다. 여기서 세 번째 가지의 길이(R3)는  $R_2 * (\text{VAR\_RED\_R})$ 가 되고 각“ θ3”는  $\theta_2 * (\text{VAR\_RED\_T})$ 가 된다.

$$x_3 = x_2 + R_3 * \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y_3 = y_2 + R_3 * \sin(\theta_1 + \theta_2 + \theta_3)$$

- 그러므로 n번째의 가지가 가지는 좌표  $x_n, y_n$ 은

$$x_n = x_{(n-1)} + R_n * \cos( (\sum \theta_{(n-1)}) + \theta_n )$$

$$y_n = y_{(n-1)} + R_n * \sin( (\sum \theta_{(n-1)}) + \theta_n )$$

가 된다.

### 3. 프랙탈 이론을 이용한 게시판 구현

#### 1) 구축환경

- CPU : celeron 266
- RAM : 96M
- OS : RedHat Linux 8.0
- Web Server : Apache 2.0
- Language : PHP 4.2.2
- DBMS : MySQL 3.23

#### 2) DB 구조

Field	Type	Key	Extra
no	int(11)	PRI	auto_increment
uid	int(11)		원글순서번호
theta	double		원점 좌표값
px	double		x 좌표값
py	double		y 좌표값
thetaN	double		자신의 진행각
theta_sum	double		진행각의 합
radiusN	double		자신의 반지름
seqno	int(11)		일련번호
depth	int(11)		깊이
indate	int(11)		입력일자
name	varchar(20)		글쓴이
passwd	varchar(20)		비밀번호
email	varchar(50)		이메일
subject	varchar(100)		글의 제목
text	text		글의 내용
counter	int(11)		조회수
p_time	double		입력소요시간

Table. 4. 프랙탈 이론을 사용하는 게시판의 DB구조

### 3) 입출력 알고리즘

#### 3-1) 원글의 입력

- ① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.
- ② 최종번호에 하나를 더한후 자신의 반지름(radiusN)을 5로 하고 진행각의 합(theta\_sum),원점좌표각(theta),자신의 진행각(thetaN)을 동일하게 초기값  $\pi/1000$ 으로 설정한다.
- ③ 깊이(depth)를 0으로 하고 레코드를 삽입한다.

#### 3-2) 답글의 입력

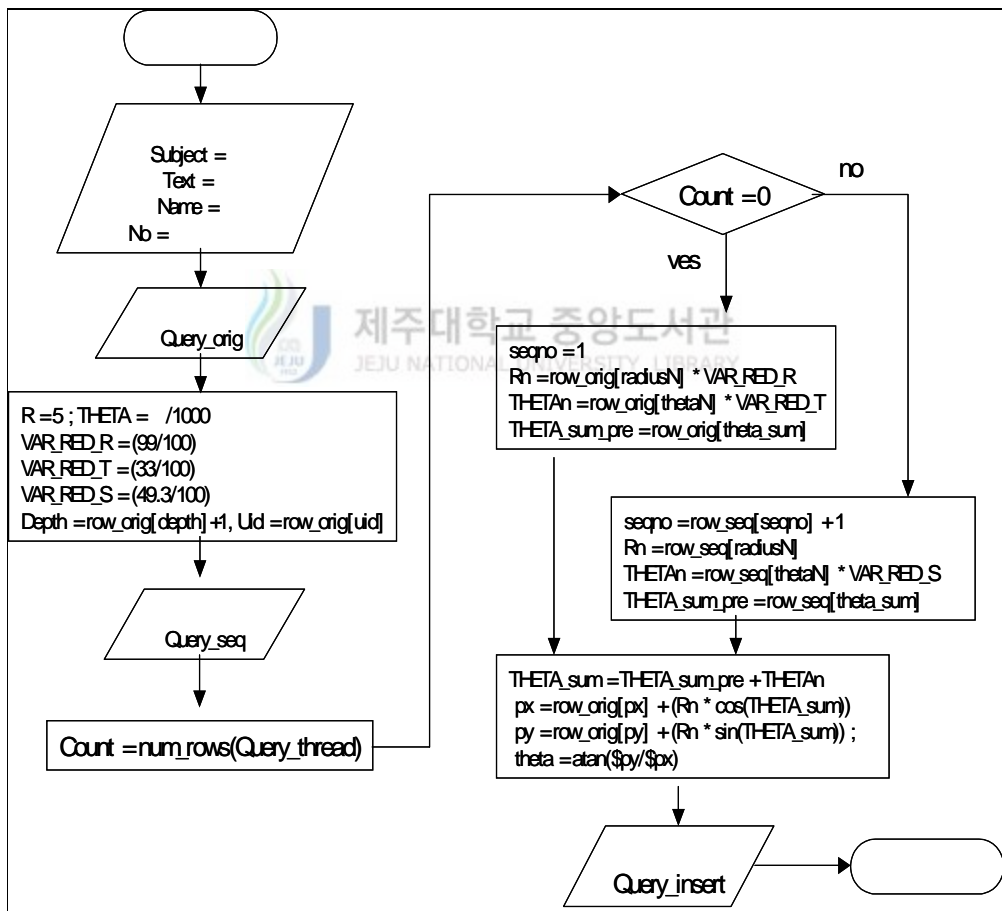


Fig. 20. 프랙탈 알고리즘을 이용한 게시판의 답글 입력순서도

- ① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.

- ② 답글과 깊이(depth)가 같으면서 일련번호가 제일 큰 글을 불러온다.
- ③ 위의 글이 없을 경우 본글에 대해서 반지름을 일정비율(99/100)로 줄이고 진행각을 일정비율(33/100)로 줄인다.
- ④ 글이 있을 경우 제일 큰 글에 대해서 반지름을 동일하게 하고 진행각을 일정비율(49.3/100)로 줄인다.
- ⑤ 자신의 반지름과 진행각을 가지고 바로 전 글의 좌표에 더하여 자신의 좌표를 계산한다.
- ⑥ 자신의 좌표를 이용하여 원점으로부터의 각을 계산해 낸다.
- ⑦ 답글을 삽입한다.

### 3-3) 게시물의 출력

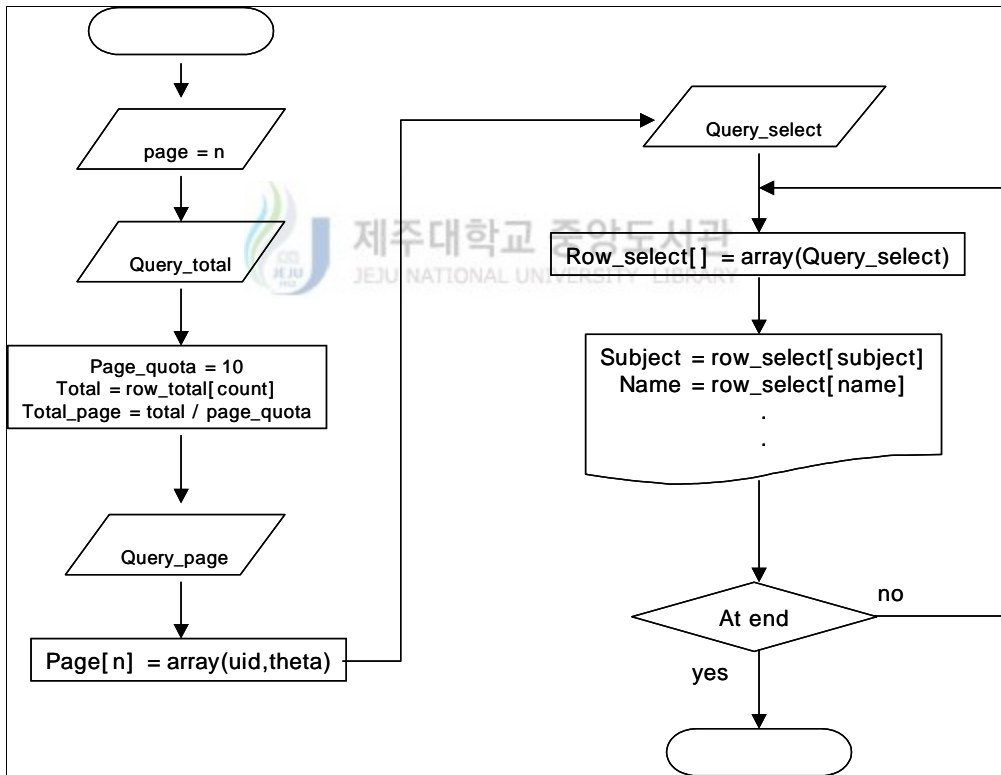


Fig. 21. 프랙탈 알고리즘을 이용한 게시판의 게시물 출력 순서도

- ① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지 당 글의 개수로 나누고 전체 페이지를 산출한다.

- ② 페이지당 작업을 위하여 원글번호(uid)의 역순과 원점좌표각(theta)순으로 원글번호와 원점좌표각을 가져온다.
- ③ 각페이지당 시작글의 원글번호와 원점좌표각을 배열에 배당한다.
- ④ 검색하고자 하는 페이지의 시작글의 원글번호와 원점좌표각을 배열에서 찾고 페이지당 글의개수 만큼 테이블에서 불러내어 출력한다.
- ⑤ 출력 질의

Query="select \* from \$table where (uid=\$a and theta>='\$b') or (uid <\$a) order by uid desc,theta limit \$page\_quota" ;

**개선알고리즘 게시판(board04)**

기본게시판 | [원글 추적하기](#) | [답글 추적하기](#) | [답글 추적하기](#)

기본게시판

no	uid	theta	depth	seano	subject	name	counter	indate	p_time
2	2	0.0031415926535898	0	0	두번째 원글	test	2	03/05/22	0.0030658437179565
3	2	0.00365795060069	1	1	제1 두번째 원글->1	test	1	03/05/22	0.009000463468718
4	2	0.0039116192717341	1	2	제1 두번째 원글->2	test	1	03/05/22	0.009698870071411
6	2	0.00422943758389	2	1	제2 두번째 원글->2-1	test	1	03/05/22	0.009679966436157
7	2	0.0043938136985406	3	1	제2 두번째 원글->2-1-1	test	0	03/05/22	0.0094799995422363
1	1	0.0031415926535898	0	0	원글 첫번째	test	0	03/05/22	0.002579829351807

/ 총 1페이지 >>
 
 / 1000 / 다음

글쓰기

리스트 출력시간 : 0.045675932965930 초

Fig. 22. 프랙탈 알고리즘을 이용한 게시판 출력모양

## IV. 결과

### 1. 테스트 전제 조건

인터넷상에서의 게시판의 입출력은 네트워크의 속도와 입,출력되는 데이터의 양에 많은 영향을 받는다. 그러나 본 논문에서 게시판의 입출력과 관련한 알고리즘의 속도 계산을 위하여 다음과 같은 전제조건을 정하고 게시판간 입출력 속도를 측정하였다.

- 1) 네트워크상의 데이터 전송속도와 무관하게 측정 하도록 한다.
- 2) 입력되는 데이터의 양은 동일하게 한다.
- 3) 에러 발생시 입력 및 출력을 중지한다.
- 4) 입력시간은 데이터를 서버로 모두 전송된 후부터 데이터베이스에 입력 직전 시간까지를 측정한다.
- 5) 출력시간은 클라이언트 환경과 무관하게 클라이언트 요청후 서버에서 클라이언트 전송직전까지의 시간을 측정한다.





## 2. 입력속도 비교

### 1) 원글 1만개 입력 시간 측정 비교 (원글 수직달기)

순성갱신형 게시판, 고정위치형 게시판, 연결리스트형 게시판과 프랙탈 알고리즘을 이용한 게시판등 모든 게시판의 원글의 입력속도와 속도변화율은 유사하게 측정되었다.

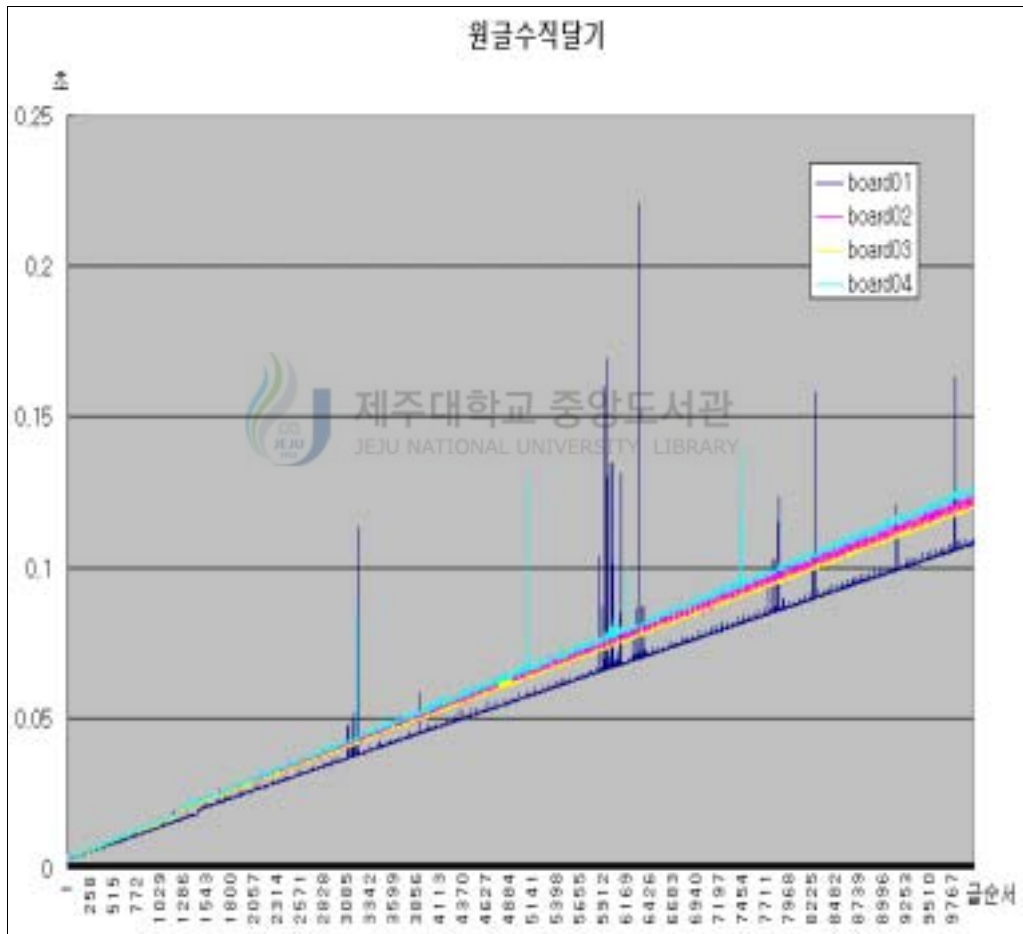


Fig. 23. 게시판간 입력속도 비교 그래프

2) 원글에 깊이 1만개의 답글 입력시간 측정 비교(답글 수평달기)

고정위치형 게시판은 문자위치고정번호가 표현할 수 있는 깊이 한계가 7개로 한정되어 입력에러가 발생하고, 또한 프랙탈을 이용한 게시판도 자신의 진행각변화가 0으로 수렴되어 668개의 답글을 달았을 경우 입력에러가 발생한다. 그 외 연결리스트형 게시판의 입력속도와 속도 변화율이 순서갱신형 게시판보다 월등히 좋을 수 있다.

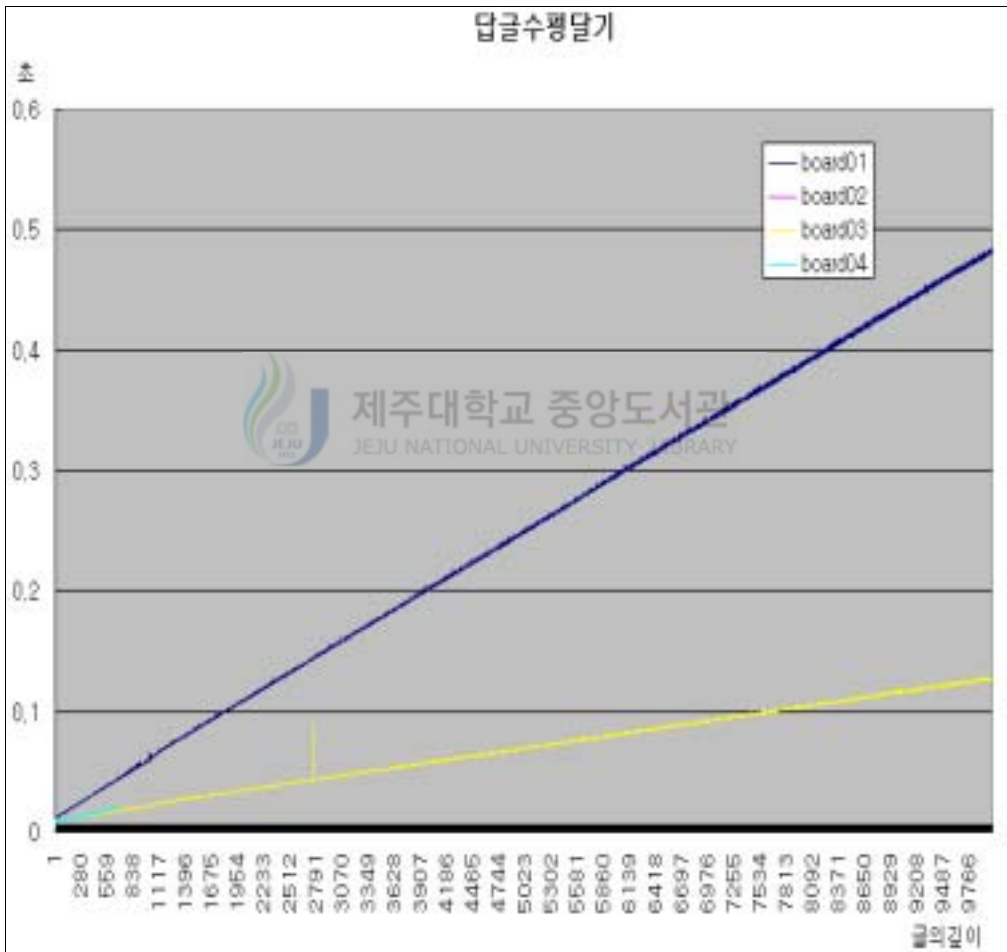


Fig. 24. 게시판간 답글 수평달기 시간 비교 그래프

### 3) 원글 1번에 1만개의 답글 입력 시간 측정 비교(답글 수직달기)

고정위치형 게시판은 답글의 개수를 표현할 수 있는 문자의 개수가 26개로 한정되어 있어 입력에러가 발생하고, 또한 프랙탈 알고리즘을 이용한 게시판의 답글 개수도 진행각의 0으로 수렴하여 48개의 답글을 달고 에러가 발생하였다. 그러나 에러발생 이전의 상태에서 비교할 경우 연결 리스트형 게시판과 동일한 속도와 속도증가율을 보이고 있으며 순서 갱신형 게시판의 입력 속도 저하가 눈에 띄게 나타나고 있다.



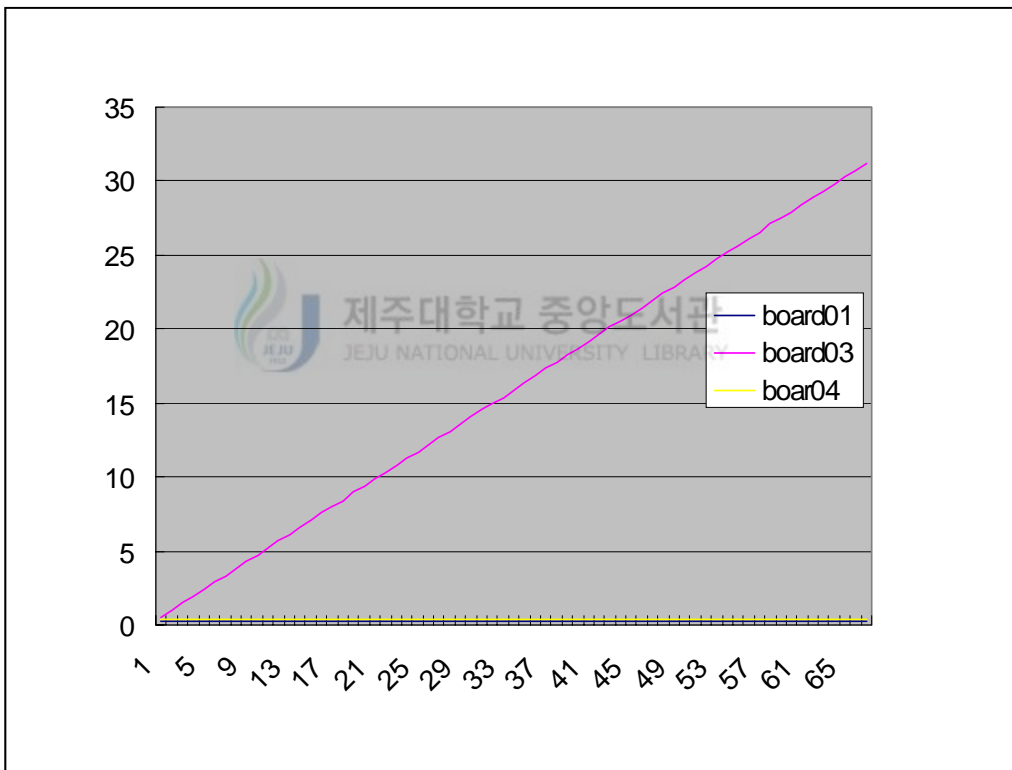
Fig. 25. 게시판간 답글 수직달기 시간 비교 그래프

### 3. 출력속도 비교

게시물의 출력은 비교를 위해서 순서갱신형과 연결리스트형의 게시물수를 프랙탈알고리즘 게시판의 개수와 같은 669개로 제한하였다.

프랙탈알고리즘 게시판과 순서갱신형 게시판은 출력속도가 일정한 반면 연결리스트형은 급격한 기울기로 출력속도가 저하되고 있다.

Fig. 26. 게시판간 리스트 출력시간 비교 그래프



## V. 결론 및 향후 연구

본 논문에서는 기존 게시판의 입출력 알고리즘과 비교하여 프랙탈 알고리즘을 이용한 게시판이 입출력에 대해서 안정된 속도를 보장하고는 있으나 답글의 깊이 제한과 개수의 제한은 역시 현재 컴퓨터 시스템상에서 극복해야 할 점으로 남아있다.

게시판간 입출력 테스트 결과를 종합하면 원글의 입력에서는 순서갱신형 게시판, 고정위치형 게시판, 연결리스트형 게시판과 프랙탈 알고리즘을 이용한 게시판 등 모든 게시판의 원글의 입력속도와 속도변화율은 유사하게 측정되었다. 원글에 깊이로 답글을 달게되는 원글 수평달기 비교에서는 고정위치형 게시판은 문자위치고정번호가 표현할 수 있는 깊이 한계가 7개로 한정되어 입력에러가 발생하고, 또한 프랙탈을 이용한 게시판도 자신의 원점각이 0으로 수렴되어 668개의 답글을 달았을 경우 입력에러가 발생한다. 그 외 연결리스트형 게시판의 입력속도와 속도 변화율이 순서갱신형 게시판보다 월등히 좋음을 알 수 있다. 원글에 답글의 개수를 달게되는 답글 수직달기는 고정위치형 게시판은 답글의 개수를 표현할 수 있는 문자의 개수가 26개로 한정되어 있어 입력에러가 발생하고, 또한 프랙탈 알고리즘을 이용한 게시판의 답글 개수도 진행각의 0으로 수렴하여 48개의 답글을 달고 에러가 발생하였다. 그러나 에러발생 이전의 상태에서 비교할 경우 연결 리스트형 게시판과 동일한 속도와 속도증가율을 보이고 있으며 순서 갱신형 게시판의 입력 속도 저하가 눈에 띄게 나타나고 있다. 게시물의 출력은 비교를 위해서 순서갱신형과 연결리스트형의 게시물수를 프랙탈알고리즘 게시판의 개수와 같은 669개로 제한하였고, 프랙탈알고리즘 게시판과 순서갱신형 게시판은 출력속도가 일정한 반면 연결리스트형은 급격한 기울기로 출력속도가 저하되고 있다.

프랙탈 알고리즘에서의 이론적 실수연산은 무한하였으나 DBMS에서의 실수형인 double은 2.225E-308의 수를 표현할 수 있고 PHP에서 원주율은 float형으로 14자리의 정밀도를 갖는다. 이로 인해 프랙탈 알고리즘을 이요한 게시판에서의 깊이와 답글이 달리는 개수가 제한이 생기며 이를 극복할수 있는 알고리즘의 확장형태가 연구되어야 하겠다.

## 참고문헌

Shishikura,M(1991), The Hausdorff dimension of the boundary of the Mandelbrot Set and Julia Set, SUNY Stony Brook, Institute for Mathematics Sciences, Preprint

Peitgen, Jurgens, Sape(1992), Fractals for the Classroom, NCTM & Springer-Verlag.

제임스 글리크(1993), 「카오스:현대과학의 대혁명」, 박배서·성운하 역, 동문사.

이석호(2003), 파일처리론 , 정익사

카일루든(2002),C로 구현한 알고리즘(Mastering Algorithms with c),허욱 역,한빛북

<http://kin.naver.com>

<http://kr.yahoo.com>

<http://www.daum.net>

[http://www.phpschool.com/bbs2/inc\\_board.html](http://www.phpschool.com/bbs2/inc_board.html)

<http://jsboard.kldp.net>

<http://www.nzeo.com>

<http://www.neoboard.net>



## 감사의 글

배움의 고품을 해결하고자 시작했던 2년 반의 대학원 생활이 마무리 단계에서 노력한다고 애는 써 봤지만 아쉬움만 남습니다. 다시 한번 도전의 기회가 주어지길 기대하면서 이 논문과 대학원 생활을 따뜻한 관심과 배려로 지도해주신 이상준 지도교수님께 깊은 존경과 감사의 마음을 드립니다.

아울러 논문 심사를 맡아 학문 연구의 진정한 의미를 일깨워주신 곽호영 교수님, 김장형 교수님, 변상용 교수님께 감사드리며, 결실을 맺기까지 항상 관심과 지도편달을 아끼지 않으신 안기중 교수님, 송왕철 교수님, 변영철 교수님께도 깊은 감사를 드립니다.

게으른 저에게 뒤처지지 않도록 충고와 조언을 아끼지 않고 많은 힘이 되어준 우리 연구실의 김영민 선생님, 박충희 선생님, 정은경씨, 금정이에게도 고마움을 전하며 같이 시작해서 어렵게 여기 까지는 정태백 선생님, 김희삼 선생님, 현병철 선생님과 행정적 절차를 도맡아 해주신 이정하 선생님에게도 고마움을 전합니다. 그리고 대학원 생활을 하며 마주친 선배님, 동기들과 후배들에게도 깊은 감사를 드립니다.

학업과 병행하느라 업무에 소홀했던 부분을 이해해 주시고 힘모아 일처리 해주신 리눅스정보기술의 이학동 사장님과, 김도균 실장, 송재경 팀장, 오지영 씨, 정현희씨, 고은하씨, 송사랑씨에게도 머리숙여 마음속 감사를 전해드리며 보다 너 나은 모습으로 노력하겠다는 마음을 전해 드립니다.

직장생활과 대학원 공부를 병행 하면서 힘들었던 시간들이었지만, 나의 부좁함까지 채워가며 늘 함께한 영원한 동반자 민숙이와 늘 보채면서도 아빠공부를 위해 많이 참아준 도희, 늘 묵묵히 지켜보면서 응원해주신 장인, 장모님과 처제, 그리고 부모님께도 감사를 드립니다.

이 글을 통해 미처 감사 드리지 못했지만, 저를 이제까지 지켜봐 주신 많은 분들께 진심으로 감사드립니다.