論文
2-2-9

확률적 이진 검색 트리 성능 추정
(Estimation of performance for random binary search trees)

김 숙 영*
(Sook-Young Kim)

요 약

이진 트리 검색에 관한 관계 모형들을 추정하고 이론 가설들을 검정하기 위하여 중복되지 않는 자연수들을 자료로 하는 3개 이상 7개 이하의 노드를 가진 모든 가능한 이진 검색 트리 들을 생성하였다. 노드 개수 별로 높이 및 균형도 에 따른 이진 검색 트리 생성 확률들을 추정하였으며 노드 개수와 트리의 높이, 검색에 필요한 비교 횟수의 관계를 나타내는 회귀 모형이 구축되었고 이진 검색 트리의 O(lg(n)) 이론이 적합도 검정 절차에 의하여 실험적으로 채택되었다.

이진 검색 트리의 균형에 따른 검색 성능의 유의적 차이들을 통계적으로 증명하기 위하여 균형도에 따라 확률적으로 생성된 이진 검색 트리들을 세 그룹으로 그룹화하고 그룹간의 검색 비교 횟수를 분산 분석 모형에 의하여 비교 분석하였다.

ABSTRACT

To estimate relational models and test the theoretical hypotheses of binary tree search algorithms, we built binary search trees with random permutations of n (number of nodes) distinct numbers, which ranged from three to seven. Probabilities for building binary search trees corresponding to each possible height and balance factor were estimated.

Regression models with variables of number of nodes, height, and average number of comparisons were estimated and the theorem of O(lg(n)) was accepted experimentally by a Lack of Test procedure.

Analysis of Variance model was applied to compare the average number of comparisons

with three groups by height and balance factor of the trees to test theoretical hypotheses of a binary search tree performance statistically.

1. 서론

최근 모든 응용 프로그램 개발에는 자료 검색 과정이 필수적으로 포함된다. 이러한 특정 자료를 검색하는 과정에는 파일이나 테이블에 저장된 레코드들의 키와 비교하여 원하는 레코드를 찾아내는 비교에 의한 방법 및 간단한 키 변환 과정을 거쳐 신속하게 검색을 수행하는 해싱에 의한 방법 등이 있다.

키 비교 방법에서는 선형으로 레코드들을 비교하여 검색할 수 있고 또한 비 선형으로 레코드를 비교하여 검색할 수도 있다.

선형으로 특정 자료를 검색하는 알고리즘은 그 구조의 단순성으로 구현은 매우 쉬우나 효율성 면에서는 검색을 수행하기 위하여 평균적으로 자료 개수에

* 정회원 : 안산공과대학 컴퓨터정보과 조교수

논문접수 : 2001. 2. 12.
심사완료 : 2001. 2. 22.

비례하는 횟수만큼 레코드들을 이동시켜야 하는 검색 속도면 에서는 비효율적으로 평가된다. 반면에 비선형적으로 자료를 검색하는 알고리즘 에서는 자료 정렬을 위한 많은 알고리즘이나 고급어를 기계어로 번역하는 컴파일 과정의 구문 분석을 위한 단계에 필수적 요소인 트리 자료 구조를 이용하여 검색할 자료를 부모 자식 노드간의 계층적 관계에 의하여 표현하고 있다. 이러한 트리는 각 노드의 서브 트리 개수에 따라 분류되며 프로그램 작성 면이나 저장 면에서 서브 트리 개수가 두 개 이하인 이진 트리로 제한하는 것이 필요하다.

이진 트리는 저장 시에도 트리 구성 요소인 노드를 연결하는 연결 필드로 인한 낭비가 서브 트리가 3개 또는 4개인 3진 또는 4진 트리 에 비하여 각각 1/6, 1/4로 줄일 수 있다는 보고가 있다 [1].

대표적인 비 선형 검색인 이진 트리 검색에서는 부모-자식 관계를 1:2로 유지함으로써 근 노드의 키 값 보다 큰 값은 왼쪽으로 작은 값은 오른쪽으로 서브 트리를 반복 형성하여 이진 검색 트리를 생성하고 있다.

따라서 검색하려는 레코드 키 값과 근 노드 값을 비교함으로써 다음에 비교할 레코드 순서가 정하여지므로 검색 속도는 트리 높이인 검색 자료 개수의 로그형에 비례한다. [2][3]. 함수에서 정의역 값이 증가함에 따라 선형 함수에 비하여 로그함수 값의 감소가 커지므로 다량의 자료 검색에서 이진 트리 검색이 선형 검색에 비하여 속도면에서 효율적인 검색이 될 수 있다.

그러나 검색할 이진 트리 구조가 불균형하게 생성되는 경우 검색 속도가 자료 수에 비례하는 비효율적인 알고리즘이 될 수 있으므로 트리의 좌우 높이가 균형을 이루도록 검색 트리를 생성함이 필요하다.

1962년 Adelson-Velskii와 Landis 는 좌·우 서브트리 높이가 1 이상 차이 나지 않을 때를 균형된 이진 트리로 제안하고 AVL 트리로 불리워지고 있다. [4]

이진 트리 검색 알고리즘 성능에 관한 이론들은 많이 존재하나 이러한 이론들을 입증할 실험 자료들은 아직 문헌에 발표되고 있지 않다.

본 연구에서는 n개 노드로 모든 가능한 이진 검색 트리 들을 확률적으로 생성하여 이진 트리 검색 성능을 실험적으로 증명하려 한다. 따라서 본 연구 목적은 다음과 같다.

첫째. 이진 검색 트리를 구성하는 노드 개수와 생성된 트리의 높이, 검색 성능의 관계 모형을 추정하고 이론적인 모형과의 적합도 검정을 수행한다.

둘째. 트리의 균형도에 따른 검색 속도를 비교 분석한다.

2. 방법

2.1. 이진 검색 트리 생성

하나의 노드는 세 개 필드(좌·우 연결 필드를 표시하는 포인터 변수 들 및 키 값)를 가진 연결 리스트(좌측 연결 링크 | 키 값 | 우측 연결 링크)에 의하여 구현되었고 하나의 트리를 구성하는 일차원 배열의 원소가 되었다. 검색 트리를 생성하기 위한 n개 노드들은 다음 알고리즘에 의하여 생성되었다 [2].

```
/* 좌, 우 연결 링크가 널인 새 노드 생성
   new=malloc(sizeof(linked_node));
   new->key= key_in[0];
   new->lchild = NULL;
   new->rchild = NULL;
/* 트리의 헤드 노드로 지정
   tree[0]  = new;
 for (i=0; i<n; i++)   {
   head=tree[0];      /* 헤드 노드
   insert_key=key_in[i+1] /* 삽입할 키 값
   current = head;    /* 이동할 노드
   pos=current;       /* 현 노드
  /* 삽입할 위치를 찾음
   while (current != NULL)          {
     pos = current;
  /* 현 노드 키 값 보다 큰 값을 삽입시
    if (insert_key > current->key)
   /*  오른쪽 연결 링크로 이동
      current = current->rchild;
  /* 현 노드 키 값 보다 작은 값을 삽입시
   if(insert_key<current->key)
  /* 왼 쪽 연결 링크로 이동
      current = current->lchild;        }
  /* 현 노드의 좌, 우 연결 링크
   if (insert_key < pos->key)
     pos->lchild = new;
    else
     pos->rchild = new;
 /* 새 노드 생성
    new=malloc(sizeof(linked_node));
    new->key = insert_key;
    new->lchild=NULL;          new->rchild=NULL;
 /* 생성된 노드를 트리에 연결
    tree[i+1]=new;
    }
```

노드 수는 3 이상 7이하로 하였다.

각 노드 수 값에 대하여 1 이상 노드 개수 이하의 자연수 들을 나열하는 모든 경우의

수 (노드 개수 !)에 해당하는 트리 구조들이 출력되었다.

각 트리 구조에서 노드들의 좌 우 연결 포인터를 비교하여 상이한 이진 트리들을 그룹화 하였다. 완성된 각 이진 검색 트리에서 근 노드들로부터 연결된 모든 노드까지 경로의 합인 내부 경로와 연결 노드가 없는 널 링크에 사각형 노드를 붙여 확장한 이진 트리의 경로의 합인 외부 경로를 측정하였다.

2.2. 검색 성능에 관한 자료 생성

이진 검색 트리에서 특정 레코드가 검색되기 위하여 검색하여야 하는 평균 원소의 개수는 내부 경로를 노드의 개수로 나눈 값에 근 노드의 수인 1을 더한 값이며 각 노드마다 두 번의 비교를 해야하고 검색이 완료된 노드에서는 한번의 비교만이 필요하므로 성공적 검색에 필요한 평균 비교 횟수는

$$(\frac{\text{내부경로길이}}{\text{노드갯수}} + 1)*2 - 1$$

식에 의하여 계산되었다. 또한 생성된 이진 검색 트리에서 원소를 찾지 못하는 경우는 근 노드로부터 잎 노드까지 하나의 노드 당 두 번의 검색이 필요하며 트리의 노드 개수 보다 하나가 많은 검색에 실패한 노드들이 확률적으로 동일하게 나타날 수 있으므로 성공하지 못한 검색의 평균 비교 횟수는 $\frac{2*\text{외부경로}}{(\text{노드갯수}+1)}$ 의 식에 의하여 계산되었다 [5].

2.3. 통계 분석

본 연구 목적으로부터 다음 연구 가설들이 유도되었다.

제 1 가설. 이진 검색 트리의 높이, 성공적 검색에 필요한 비교 횟수, 비 성공적 검색에 필요한 비교 횟수는 노드의 개수에 대하여 로그 함수를 이루는가?

제 2 가설. 균형된 이진 트리, 비균형된 이진 트리 및 경사진 이진 트리 사이에 검색 속도에 차이가 존재하는가? 이다.

제 1 가설 검정을 위하여 설정된 통계 영 가설은 $Y(n) = \beta_0 + \beta_1 \cdot \log_2(n)$이다.

제 2 가설 검정을 위하여 설정된 통계 모형은 $Y(n) = \mu + \alpha_i + \tau_j + \varepsilon_{ij}$ (α_i: 균형 그룹, τ_j: 트리 높이) 이고 검정할 영가설은 $\alpha_i = 0$이다.

제 1 가설은 회귀 모형에서의 적합도 검정(Lack of Fit test)에 의하여, 제 2 가설은 트리의 균형도를 주요인으로, 트리의 높이를 블록으로 처리하는 블록화된 분산 분석법에 의하여 검정되었으며 가설 기각 시 그룹끼리의 차이 유무 비교를 위하여 Tukey의 다중 비교법을 적용한다.

모든 수치적 계산 결과들은 SAS(Statistical Analysis System)의 subroutine 들을 이용한 프로그램 실행 결과들로 얻어졌다. 참인 영가설을 기각할 오류를 범할 오류의 최대 허용 한계 5%를 택하였다 [6][7].

3. 결과

3.1. 이진 검색 트리 생성 확률

n 개 노드를 가진 상이한 이진 검색 트리들의 높이에 따른 분포가 <표 1>에 기술되어 있다.

노드의 개수가 많아질수록 한쪽으로만 생성되는 사향 트리가 될 확률은 감소하며 높이가 「노드 개수의 중위수」인 트리가 생성될 확률이 가장 높았다.

이론 상 n 개 노드로부터 생성되는 상이한 이진 트리의 수는 서브 트리들을 결합 법칙이 성립되는 n 개 행렬을 곱하는 방법의 수와 같은 논리에 의하여 $b_n = \sum_{i=1}^{n-1} b_i b_{n-i}$ ($b_0 = 1$, $n>1$)의 공식을 순환식과 이항 정리를 이용하여 변환한 $\binom{2n}{n}\frac{1}{n+1}$ 의 식에 노드의 개수를 n에 대입하여 계산된 값 ($b_3 = 5$, $b_4 = 14$, $b_5 = 42$, $b_6 = 132$, $b_7 = 429$)이며 <표 1>의 결과와 일치하였다 [5].

<표 1> 상이한 이진 검색 트리의 높이별 빈도

<Table.1> Frequency of distinct binary search trees by height

높이 / 노드수	1	2	3	4	5	6	상이한 트리 수
3	1 (20%)	4 (80%)	0	0	0	0	5 (100%)
4	0	6 (42.9%)	8 (57.1%)	0	0	0	14 (100%)
5	0	6 (14.3%)	20 (47.6%)	16 (38.1%)	0	0	42 (100%)
6	0	4 (3.0%)	41 (31.1%)	55 (41.7%)	32 (24.2%)	0	132 (100%)
7	0	1 (0.2%)	68 (15.9%)	153 (35.7%)	143 (33.3%)	64 (14.9%)	429 (100%)

노드 개수에 따른 트리 높이 및 균형 인수의 분포가 <표 2>(a) 에서 <표 2>(e)에 기술되어 있다.

4개 이하의 노드를 가진 트리 에서는 생성된 모든 이진 검색 트리의 50% 이상이 균형을 이루었다. 노드 수가 증가함에 따라 트리가 균형을 이룰 확률은 감소하여 7개 노드를 가진 트리에서는 생성된 이진 검색 트리의 16% 만이 균형을 이루었다.

<표 2> (a) 3개 노드를 가진 이진 검색 트리의 높이와 균형 인수에 따른 도표화

<Table 2>(a) Tabulation of Height and Balance Factor for binary search trees with three nodes

높이 / 균형인 수	0	2	합
1	2	0	2(33.3%)
2	0	4	4(66.7%)
합	2(33.3%)	4(66.7%)	6(100%)

<표 2>(b) 4개 노드를 가진 이진 검색 트리의 높이와 균형 인수에 따른 도표화

<Table 2>(b) Tabulation of Height and Balance Factor for binary search trees with four nodes

균형인수 / 높이	1	2	3	합
2	12	4	0	16 (66.7%)
3	0	0	8	8 (33.3%)
합	12(50%)	4(16.7%)	8(33.3%)	24(100%)

<표 2>(c) 5개 노드를 가진 이진 검색 트리의 높이와 균형 인수에 의한 도표화

<Table 2>(c) Tabulation of Height and Balance Factor for binary search trees with five nodes

높이 \ 균형인 수	1	2	3	4	합
2	46	0	0	0	46 (38.3%)
3	0	26	32	0	58 (48.3%)
4	0	0	0	16	16 (13.3%)
합	46 (38.3%)	26 (21.7%)	32 (26.7%)	16 (13.3%)	120(100%)

<표 2>(d) 6개 노드를 가진 이진 검색 트리의 높이와 균형 인수에 의한 도표화

<Table 2>(d) Tabulation of Height and Balance Factor for binary search trees with six nodes

높이 \ 균형인 수	1	2	3	4	5	합
2	80	0	0	0	0	80(11.1%)
3	0	320	84	0	0	404(56.1%)
4	0	0	80	124	0	204(28.3%)
5	0	0	0	0	32	32(4.4%)
합	80 (11.1%)	320 (44.4%)	164 (22.8%)	124 (17.2%)	32 (4.4%)	720 (100%)

<표 2>(e) 7개 노드를 가진 이진 검색 트리의 높이와 균형 인수에 의한 도표화

<Table 2>(e) Tabulation of Height and Balance Factor for binary search trees with seven nodes

높이 \ 균형인 수	0	1	2	3	4	5	6	합
2	80	0	0	0	0	0	0	80 (1.6%)
3	0	756	1276	208	0	0	0	2240 (44.4%)
4	0	0	0	1248	804	0	0	2052 (40.7%)
5	0	0	0	0	192	412	0	604 (12.0%)
6	0	0	0	0	0	0	64	64 (2.3%)
합	80 (1.6%)	756 (15.0%)	1256 (24.9%)	1456 (28.9%)	996 (19.8%)	412 (8.2%)	64 (1.3%)	5040 (100%)

3.2. 이진 검색 트리의 성능 추정

생성된 이진 검색 트리의 높이 및 검색을 위한 평균 비교 횟수들과 노드 개수의 관계가 [그림 1]에 그래프화 되어 있다. 세 곡선 모두가 로그형을 이루고 있었고 회귀 모형의 적합도 검정인 Lack of Fit test 결과 로그 모형의 가설이 채택되었다($p>0.1$)[6].

추정된 회귀 모형을 이진 트리 이론 분석에 필요한 밑(base)이 2인 로그 함수로 변환하면 노드 개수 (n) 과 생성되는 이진 트리의 높이 관계는 $H(n)=1.62 \cdot \lg(n)-0.92$,

성공적 검색에 필요한 비교 횟수와의 관계는 $S(n)=1.69 \cdot \lg(n)+0.06$ 이며 비 성공적 검색에 필요한 비교 횟수와의 관계는 $U(n)=2.07 \cdot \lg(n)+1.00$이었다.

세 개의 회귀 모형에서 R-square 값은 0.99 이상으로 노드 개수와 이진 검색 트리 성능 관계는 추정된 회귀 모형에 의하여 99% 이상이 설명될 수 있는 매우 설명력이 높은 모형으로 추정되었다.

3.3. 트리 균형도에 따른 검색 성능 비교

확률적으로 생성된 이진 검색 트리들을 트리 높이와 균형 인수에 따라 균형 인수가 1이하인 경우를 균형 트리, 트리 높이가 (노드 개수-2) 이상이고 균형 인수가 트리 높이인 경우를 사향 트리, 균형이 아닌 트리 중 경사가 아닌 경우를 불균형 트리로 세 그룹으로 그룹화 하였다. 각 그룹마다 검색에 성공하였을 때와 실패하였을 때 필요한 비교 횟수들이 <표 3>에 기술되어 있다.

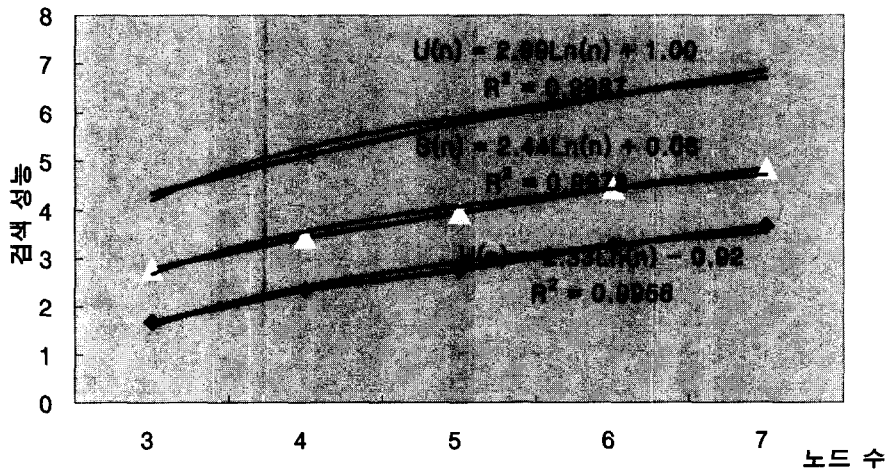
<표 3> 균형도에 따른 비교 횟수(평균 ± 표준편차)

<Table 3> Average number of comparison of search trees by balance (Mean±S.D.)

균형도 / 비교횟수	균형된 트리 (n=976)	불균형 트리 (n=4274)	사향 트리 (n=660)
성공 검색	4.04± 0.26a	4.76±0.49b	5.97±0.64c
비성공 검색	6.13±0.30a	4.77±0.45b	7.80±0.65c

Values with different superscripts are statistically significantly different ($p<0.01$)

검색에 필요한 비교 횟수는 노드 개수에 따라 차이가 있으므로 노드 개수를 블록으로 설정한 후 블록화 된 일원 분산 분석 모형에 의하여 가설 검정하였다. 검색 성공 여부에 관계없이 필요한 평균 비교 횟수는 트리의 균형도에 따라 유의적인 차이가 존재하였다($p<0.01$). 그룹끼리의 비교를 위한 다중 비교 결과 AVL 트리의 검색의 경우 비교 횟수가 가장 적었고 사향 트리의 비교 횟수가 가장 증가되었다($p<0.01$).



[그림 1] 노드 수와 검색 성능의 함수 관계

[Fig.1] Functional Relations of number of nodes and performance of binary search trees

4. 고찰

이론 상 이진 검색 트리는 검색 성능이 로그형으로 선형 검색에 비하여 우수한 검색 알고리즘으로 평가되나 검색할 자료의 분포에 따라 검색 트리가 불균형하게 생성되는 경우 성능이 저하 될 수도 있다. 따라서 임의의 자료에 이진 검색 트리 알고리즘을 적용할 때 생성되는 이진 검색 트리의 높이와 균형 인수를 확률적으로 추정함이 필요하다.

본 연구에서는 1부터 시작하여 노드 개수만큼의 자연수들을 검색 자료로 순열 상 모든 가능한 경우의 수에 대하여 이진 검색 트리들을 생성하여 트리 균형에 따른 확률들을 추정하였으며 로그형인 검색 성능을 통계적으로 증명하였다.

3개 이상 7개 이하의 노드를 가진 이진 검색 트리 모양은 트리 높이가 노드 개수의 중위수로 생성될 확률이 가장 높았다.

모든 가능한 자료로부터 얻어진 이진 검색 트리에서 균형된 AVL 트리 및 사향 트리가 될 확률은 노드 개수가 증가함에 따라 감소하여 7개 노드로 생성된 이진 검색 트리에서 16%가 AVL 트리로, 1% 정도가 사향 트리로 생성되었다.

노드 개수에 따른 이진 검색 트리 높이와 검색에 필요한 비교 횟수에 대한 O(lg(n)) 이론이 적합도 검정 (Lack of Fit test) 결과 채택되었다.

노드 개수와 검색 성능에 관한 회귀 모형 추정 결과 $O(n) \approx C \cdot \lg(n)$ 으로 표현될 수 있으며 트리 높이 에서는 C=1.62, 성공적 검색의 비교 횟수에서는 C=1.69, 비 성공적 검색 비교 횟수에서는 C=2.07 이었다.

완전 균형된 이진 트리에서는 $C \approx 1$로 기대되므로 일반적으로 이진 검색 트리 성능은 최적 조건에 비하여 60% 정도 저하되는 것으로 예상할 수 있다.

Kruse 등의 이진 검색 트리에서의 평균 비교 횟수는 완전 균형이진 트리 경우 보다 39% 높다는 결과에 비하여 본 실험 결과는 과다 추정되었다.

노드의 수를 3개 이상 7개 이하로 제한된 실험 조건에 의한 차이로 설명될 수 있을 것이다 [8][9][10].

본 실험 결과로 노드 개수가 7개 이하일 때 이진 검색 트리의 검색 성능이 로그형임이 실험적으로 증명되었다.

함수 이론 상 정의역 값이 증가함에 따라 선형에 비하여 로그형 함수값의 감소가 심해지므로 이진 트리 검색의 성능은 더욱 향상되리라 예측된다.

그러므로 높이가 4 이상인 완전 균형이진 트리를 생성할 수 있도록 30개 이상의 노드를 가진 이진 검색 트리로 본 실험이 확장되면 트리 균형성이 검색 성능에 미치는 통계 분석에서 높이와 균형 인수를 연속 변수로 처리하여 검색 성능이 저하되는 변화점을 찾을 수 있을 것이다.

또한 자료의 통계 분포에 따른 이진 검색 트리 적합도에 관한 가설들을 유도하고 추정하는 시도 또한 필요할 것이다.

※ 참고문헌

[1] 박영근. 자료 구조 기초와 활용 21세기사, p.255-280, 1998.

[2] 강맹규. 자료 구조 홍릉과학 출판사 p.387-406, 1997.

[3] R. Lesuisse. Some lessons drawn from the history of the binary search algorithm. The Computer Journal, 26, pp.154-163, 1983.

[4] 임형근. C 언어로 구현한 자료 구조론, 기술 연구사 p.326-352, 1996.

[5] 이석호 역. C로 쓴 자료 구조론. 회중당, p.242-256, 1993.

[6] J Neter, W Wasserman and M H. Kutner, "Applied Linear Regression Models". Richard D. Irwin Inc., pp.130-140, 1988.

[7] S Glantz and B Slinker, "Primer of applied regression and analysis of variance", McGraw-Hill Inc., pp. 292-295, 1990.

[8] T Cormen, C Leiserson and R Rivest, "Introduction to Algorithms", McGraw-Hill, pp. 254-262, 1990.

[9] R. Kruse, B Leung and C Tondo; "Data structures and program design in C", Prentice-Hall Inc., pp. 327-330, 1991.

[10] H Chang and S Iyengar, Efficient algorithms to globally balance a binary search tree. Communications of the ACM, Vol 27, pp. 695-702, 1984.

김 숙 영

1990년 미국 오하이오 주립대학교 컴퓨터 과학과 졸업 (이학사)
미국 오하이오 주립대학교 대학원 통계학과 졸업 (응용 통계학 석사)
현재 안산 공과 대학 컴퓨터 정보과 조교수
관심 분야 : 전산 수학, 전산 통계