

Optimal Binary Search Tree Algorithm

소프트웨어학부
20170294 박해영

< 목 차 >

1. 목표	-----	3p
2. Problem & Input/Output	-----	3p
3. 구현 Language & 사용 Tool	-----	3p
4. 교재의 입력 데이터 테스트	-----	4p
5. 자작 입력 데이터 생성 & 알고리즘 과정 손계산	-----	5p
6. 자작 입력 데이터 테스트	-----	7p

1. 목표

이진검색 트리(Binary search tree)에서 아이템들을 정리하는 최적의 방법을 결정하는 알고리즘을 만들기 위해 트리구조에 대한 전반적인 조건들을 이해한다. 이진 트리의 검색 과정에서 키를 찾기 위해 프로시저 search 가 수행한 비교 횟수(검색 시간)와 키가 검색 키가 될 확률 값을 통해 트리를 어떻게 구성해야 하는지 최적의 원칙에 따라 동적계획 알고리즘 구현 방법을 통해 이해하고자 한다.

2. Problem & Input / Output

* **Problem** : 키의 집합에 대한 최적 이진검색 트리를 구하라.

여기서 각 키에 대해서 검색키가 될 확률이 주어진다.

* **Input** : n (키의 개수), p (실수 배열), 여기서 $p[i]$ 는 $(i-1)$ 는 i 번째 키를 찾을 확률이다.

* **Output** : 변수 minavg (최적 이진검색 트리에 대한 평균 검색시간),

최적 트리를 구축할 수 있는 2차원 배열 R ,

최적 트리를 구축하기 위해 계산한 검색 시간이 담긴 2차원 배열

3. 구현 Language & 사용 Tool

* **구현 언어** : C language

* **사용 Tool** : Visual Studio 2017

4. 교재의 입력 데이터 테스트

1) Ex3.9

```
Input of Tree size : 4
1번째 key of tree : Don
1번째 Probability of tree : 0.375
2번째 key of tree : Isabelle
2번째 Probability of tree : 0.375
3번째 key of tree : Ralph
3번째 Probability of tree : 0.125
4번째 key of tree : Wally
4번째 Probability of tree : 0.125

@@@   A array   @@@

0.000 0.000 0.000 0.000 0.000
0.000 0.375 1.125 1.375 1.750
0.000 0.000 0.375 0.625 1.000
0.000 0.000 0.000 0.125 0.375
0.000 0.000 0.000 0.000 0.125
0.000 0.000 0.000 0.000 0.000

@@@   R array   @@@

0 0 0 0 0
0 1 1 2 2
0 0 2 2 2
0 0 0 3 3
0 0 0 0 4
0 0 0 0 0

Total Search Time of Optimal binary tree : 1.750000
```

5. 자작 입력 데이터 생성 & 알고리즘의 과정 손계산

자작데이터 생성 $\Rightarrow n=5$

$$P[] = \{0.1, 0.2, 0.3, 0.2, 0.2\}$$

A	0	1	2	3	4	5
1	0	0.1	0	0	0	0
2	0	0	0.2	0	0	0
3	0	0	0	0.3	0	0
4	0	0	0	0	0.2	0
5	0	0	0	0	0	0.2
6	0	0	0	0	0	0

R	0	1	2	3	4	5
1	0	1	0	0	0	0
2	0	0	2	0	0	0
3	0	0	0	3	0	0
4	0	0	0	0	4	0
5	0	0	0	0	0	5
6	0	0	0	0	0	0

* 초기 배열 형성.
 $A[i][i] = P[i-1], R[i][i] = i$

Array A & R 채우기.

① Diagonal = 1

$$1) \bar{i} = 1, \bar{j} = 1+1=2.$$

pro = 0.3 // Prob 값을 통해 확률값을 세기.
 $(\bar{i}-1 \sim \bar{j}-1)$ 까지의 합.

* k=1 일때

$$A[1][2] == 0 \parallel A[1][0] + A[2][2] < A[1][2] \Rightarrow \text{true}$$

$$A[1][2] = A[1][0] + A[2][2] = 0 + 0.2 = 0.2.$$

$$R[1][2] = 1.$$

* k=2 일때

$$A[1][2] == 0 \parallel A[1][1] + A[2][2] < A[1][2] \Rightarrow \text{true}$$

$$A[1][2] = A[1][1] + A[2][2] = 0.1 + 0 = 0.1$$

$$R[1][2] = 2.$$

$$A[1][2] += \text{pro}; \therefore A[1][2] = 0.4.$$

$$2) \bar{i} = 2, \bar{j} = 2+1=3.$$

pro = 0.5 // $P[1] \sim P[2]$ 까지의 합

* k=2 일때

$$A[2][3] == 0 \parallel A[2][1] + A[3][3] < A[2][3] \Rightarrow \text{true}$$

$$A[2][3] = A[2][1] + A[3][3] = 0 + 0.3 = 0.3$$

$$R[2][3] = 2.$$

* k=3 일때

$$A[2][3] == 0 \parallel A[2][2] + A[4][3] < A[2][3] \Rightarrow \text{true}$$

$$A[2][3] = A[2][2] + A[4][3] = 0.2 + 0 = 0.2$$

$$R[2][3] = 3.$$

$$A[2][3] += \text{pro}; \therefore A[2][3] = 0.7.$$

$$3) \bar{i} = 3, \bar{j} = 3+1=4.$$

pro = 0.5 // $P[2] \sim P[3]$ 까지의 합.

* k=3 일때

$$A[3][4] == 0 \parallel A[3][2] + A[4][4] < A[3][4] \Rightarrow \text{true}$$

$$A[3][4] = A[3][2] + A[4][4] = 0 + 0.2 = 0.2$$

$$R[3][4] = 3.$$

* k=4 일때

$$A[3][4] == 0 \parallel A[3][3] + A[5][4] < A[3][4] \Rightarrow \text{false}$$

$$A[3][4] += \text{pro}; \therefore A[3][4] = 0.7$$

4) $i=4, j=4+1=5$

$pro = 0.4$ // $P[3] \sim P[4]$ 까지의 합

* $k=4$ 일때

$A[4][5] == 0 \parallel A[4][3] + A[5][5] < A[4][5]$
 $\Rightarrow true$

$A[4][5] = A[4][3] + A[5][5] = 0 + 0.2 = 0.2$

$R[4][5] = 4$

* $k=5$ 일때

$A[4][5] == 0 \parallel A[4][4] + A[5][5] < A[4][5]$
 $\Rightarrow false$

$A[4][5] += pro ; \therefore A[4][5] = 0.6$

(A)

	0	1	2	3	4	5
1	0	0.1	0.4	0	0	0
2	0	0	0.2	0.7	0	0
3	0	0	0	0.3	0.7	0
4	0	0	0	0	0.2	0.6
5	0	0	0	0	0	0.2
6	0	0	0	0	0	0

(R)

	0	1	2	3	4	5
1	0	1	2	0	0	0
2	0	0	2	3	0	0
3	0	0	0	3	3	0
4	0	0	0	0	4	4
5	0	0	0	0	0	5
6	0	0	0	0	0	0

$\Rightarrow diagonal = 1$ 일때 결과물

② $diagonal = 2$

1) $i=1, j=3, pro = 0.6$

* $k=1 \sim 3$

최대값 $\Rightarrow k=2$ 일때

$A[1][3] = A[1][1] + A[3][3] = 0.4$

$R[1][3] = 2$

$A[1][3] += pro ; \therefore A[1][3] = 1.0$

2) $i=2, j=4, pro = 0.7$

* $k=2 \sim 4$

최대값 $\Rightarrow k=3$

$A[2][4] = A[2][2] + A[4][4] = 0.4$

$R[2][4] = 3$

$A[2][4] += pro ; \therefore A[2][4] = 1.1$

3) $i=3, j=5, pro = 0.7$

* $k=3 \sim 5$

최대값 $\Rightarrow k=4$

$A[3][5] = A[3][3] + A[5][5] = 0.5$

$R[3][5] = 4$

$A[3][5] += pro ; \therefore A[3][5] = 1.2$

(A)

	0	1	2	3	4	5
1	0	0.1	0.4	1.0	0	0
2	0	0	0.2	0.7	1.1	0
3	0	0	0	0.3	0.7	1.2
4	0	0	0	0	0.2	0.6
5	0	0	0	0	0	0.2
6	0	0	0	0	0	0

(R)

	0	1	2	3	4	5
1	0	1	2	2	0	0
2	0	0	2	3	3	0
3	0	0	0	3	3	4
4	0	0	0	0	4	4
5	0	0	0	0	0	5
6	0	0	0	0	0	0

③ diagonal = 3.

1) $i=1, j=4, \text{pro} = 0.8.$

* $k=1 \sim 4.$

최대값 $\Rightarrow k=3.$

$$A[1][4] = A[1][2] + A[4][4] = 0.6$$

$$R[1][4] = 3.$$

$$A[1][4] += \text{pro}; \quad A[1][4] = 1.4.$$

2) $i=2, j=5, \text{pro} = 0.9.$

* $k=2 \sim 5$

최대값 $\Rightarrow k=3.$

$$A[2][5] = A[2][2] + A[4][5] = 0.8.$$

$$R[2][5] = 3.$$

$$A[2][5] += \text{pro}; \quad A[2][5] = 1.7.$$

(A)	0	1	2	3	4	5
1	0	0.1	0.4	1.0	1.4	0
2	0	0	0.2	0.7	1.1	1.7
3	0	0	0	0.3	0.7	1.2
4	0	0	0	0	0.2	0.6
5	0	0	0	0	0	0.2
6	0	0	0	0	0	0

(R)	0	1	2	3	4	5
1	0	1	2	2	3	0
2	0	0	2	3	3	3
3	0	0	0	3	3	4
4	0	0	0	0	4	4
5	0	0	0	0	0	5
6	0	0	0	0	0	0

$\Rightarrow \text{diagonal} = 3$ 의 결과를

④ diagonal = 4.

1) $i=1, j=5, \text{pro} = 1.0.$

* $k=1 \sim 5.$

최대값 $\Rightarrow k=3.$

$$A[1][5] = A[1][2] + A[4][5] = 1.0.$$

$$R[1][5] = 3.$$

$$A[1][5] += \text{pro}; \quad A[1][5] = 2.0.$$

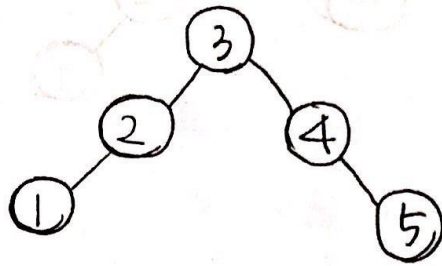
(A)	0	1	2	3	4	5
1	0	0.1	0.4	1.0	1.4	2.0
2	0	0	0.2	0.7	1.1	1.7
3	0	0	0	0.3	0.7	1.2
4	0	0	0	0	0.2	0.6
5	0	0	0	0	0	0.2
6	0	0	0	0	0	0

(R)	0	1	2	3	4	5
1	0	1	2	2	3	3
2	0	0	2	3	3	3
3	0	0	0	3	3	4
4	0	0	0	0	4	4
5	0	0	0	0	0	5
6	0	0	0	0	0	0

$\Rightarrow \text{diagonal} = 4$ 의 결과를
최종 결과물

Total Search time
of optimal binary tree
= 2.0.

* 최적 이진 검색 트리.



6. 자작 입력 데이터 테스트

```

Input of Tree size : 5
1번째 key of tree : Don
1번째 Probability of tree : 0.1
2번째 key of tree : Isabelle
2번째 Probability of tree : 0.2
3번째 key of tree : Ralph
3번째 Probability of tree : 0.3
4번째 key of tree : Wally
4번째 Probability of tree : 0.2
5번째 key of tree : James
5번째 Probability of tree : 0.2

@@@   A array   @@@

0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.100 0.400 1.000 1.400 2.000
0.000 0.000 0.200 0.700 1.100 1.700
0.000 0.000 0.000 0.300 0.700 1.200
0.000 0.000 0.000 0.000 0.200 0.600
0.000 0.000 0.000 0.000 0.000 0.200
0.000 0.000 0.000 0.000 0.000 0.000

@@@   R array   @@@

0 0 0 0 0 0
0 1 2 2 3 3
0 0 2 3 3 3
0 0 0 3 3 4
0 0 0 0 4 4
0 0 0 0 0 5
0 0 0 0 0 0

Total Search Time of Optimal binary tree : 2.000000
  
```