Suppose that we have n records, a_i, $i = 1,\ldots,n$, and these n records are stored in the nodes in a binary search tree. We call this kind of node that data node and each node (record) is associated with an access probability p_i. If a search in the binary search tree reaches an external node between a_i and a_{i+1}, we say that the search reaches a failure node. There are n+1 failure nodes. Each failure node is associated with a probability q_i, $i = 0,\ldots,n$. A node (data node or failure node) contributes cost $p \cdot h$ to the total search cost where p is the associated probability and h is the depth of the node. The binary search tree stores these n records is an optimal binary search tree if the total cost ($\sum_i p_i h_i + \sum_j q_j h_j$) is the least.

Which two of the following statements are true.

(a) Suppose there are 4 records with key values (10, 15, 20, 25), p_i are (3/16, 3/16, 1/16, 1/16), and q_i are (2/16, 3/16, 1/16, 1/16, 1/16), the optimal binary search tree is as shown in Figure 3-2.

(b) Suppose there are 4 records with key values (10, 15, 20, 25), p_i are (3/16, 3/16, 1/16, 1/16), and q_i are (2/16, 3/16, 1/16, 1/16, 1/16), the optimal binary search tree is as shown in Figure 3-3.

(c) The optimal binary search is constructed by using the divide and conquer technique that can be done in O(nlogn) time.

(d) If there are n records and every node has the identical aceess probability, the the cost for the optimal binary is Θ(nlogn).

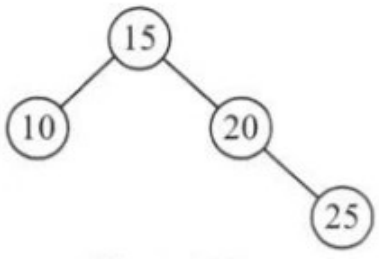


Figure 3-2

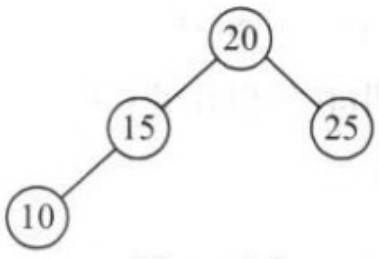


Figure 3-3

【98 年交大資工所】

解 (a),(d)

Ans.

2022.4.5
花 7:35

	0	1	2	3	4
N		(10)	(15)	(20)	(25)
P		$\frac{3}{16}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
q	$\frac{2}{16}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$

w	0	1	2	3	4
1	2	8	12	14	16
2		3	7	9	11
3			1	3	5
4				1	3
5					1

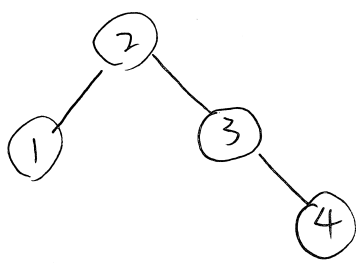
e	0	1	2	3	4
1	2	13	25	32	40
2		3	11	17	25
3			1	5	11
4				1	5
5					1

r	1	2	3	4
1	1	1	2	2
2		2	2	2
3			3	3
4				4

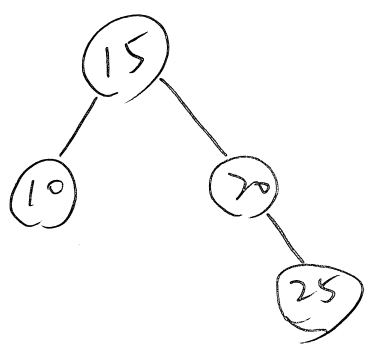
r[1.4] = 2

r[3.4] = 3

⇒



⇒



(a) 是對的

(d) 是對的，平衡的樹，有 n 個點，depth = $\log n$

⇒ $n \log n$