

The BST and Hash data generated when part of the given test data is not operated are as follows

APP1

Cat	NodeCount	TreeDepth	LeftDepth	RightDepth
Games	4	3	2	0

Hash 0	1	Hash 6	1	Hash 8	1
Hash 9	1				

App4

Cat	NodeCount	TreeDepth	LeftDepth	RightDepth
Books	1	1	0	0
Business	2	2	1	0
Catalogs	1	1	0	0
Education	3	3	2	0
Entertainment	2	2	0	1
Finance	3	3	0	2
Food & Drink	1	1	0	0

Hash 0	1	Hash 3	1	Hash 9	1
Hash 13	2	Hash 14	1	Hash 15	1
Hash 17	1	Hash 18	1	Hash 22	1
Hash 23	1	Hash 24	1	Hash 27	1

App5

Cat	NodeCount	TreeDepth	LeftDepth	RightDepth
Books	1	1	0	0
Business	2	2	1	0
Catalogs	1	1	0	0
Education	3	3	2	0
Entertainment	2	2	0	1
Finance	3	3	0	2
Food & Drink	1	1	0	0
Game	2	2	0	1
Health & Fitness	4	3	2	1
Kids	2	2	1	0
Lifestyle	2	2	1	0
Magazines & Newspapers	2	2	1	0
Medica	2	2	1	0

Music	2	2	0	1
Navigation	2	2	1	0
News	1	1	0	0
Photo & Video	2	2	0	1
Productivity	1	1	0	0
Reference	1	1	0	0
Social Networking	2	2	1	0
Sports	2	2	1	0
Travel	2	2	0	1
Utilities	3	3	0	2
Weather	1	1	0	0

Hash 1	2	Hash 2	1	Hash 4	1
Hash 5	1	Hash 7	1	Hash 8	1
Hash 10	1	Hash 12	1	Hash 13	1
Hash 14	1	Hash 16	1	Hash 18	1
Hash 20	1	Hash 23	1	Hash 30	1
Hash 32	3	Hash 34	2	Hash 35	1
Hash 36	1	Hash 38	1	Hash 39	1
Hash 43	1	Hash 49	1	Hash 51	1
Hash 55	1	Hash 59	1	Hash 61	3
Hash 66	1	Hash 67	1	Hash 68	1
Hash 69	1	Hash 70	1	Hash 79	1
Hash 80	1	Hash 85	1	Hash 86	1
Hash 88	1	Hash 90	1	Hash 91	2

App8

Cat	NodeCount	TreeDepth	LeftDepth	RightDepth
Weather	4	3	1	2
Sports	12	6	5	3

Hash 0	1	Hash 2	1	Hash 5	1
Hash 11	1	Hash 14	1	Hash 16	1
Hash 21	1	Hash 22	1	Hash 27	1
Hash 29	1	Hash 30	1	Hash 31	1
Hash 34	1	Hash 35	1	Hash 36	2

- (1) From the above data, it can be seen that the hash function can distribute apps to different locations by name, and there are only a few locations where there are multiple records. Therefore, it can be considered that this hash function is efficient.
- (2) It can be seen from the above data that most binary trees are not balanced. And this is only in the memory allocation stage, if it is processed according to the given operation, it may increase the imbalance of the binary tree. If the binary tree is unbalanced, the worst will degenerate into a linked list, then its query complexity may become $O(n)$, which greatly reduces the query efficiency and loses its meaning.
- (3) Due to the limited test data, the test results are too few to fully prove that Hash is more friendly to the find operation. But theoretically, the query complexity of Hash can reach $O(1)$, while the binary tree is $O(\log n)$. In contrast, Hash is more suitable for query.