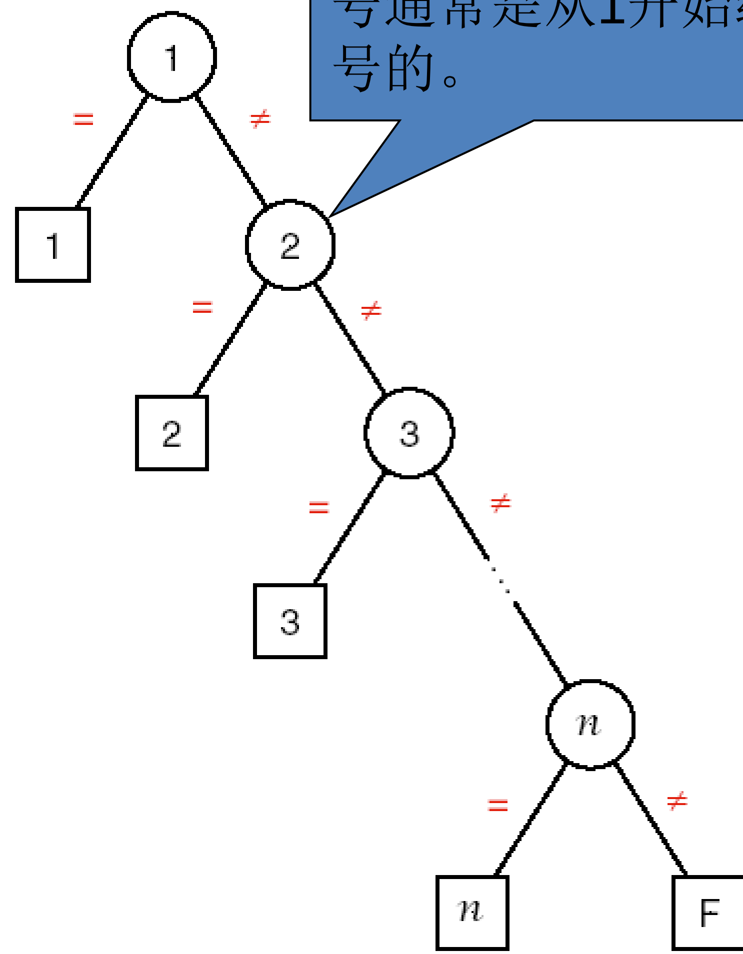
Comparison Trees

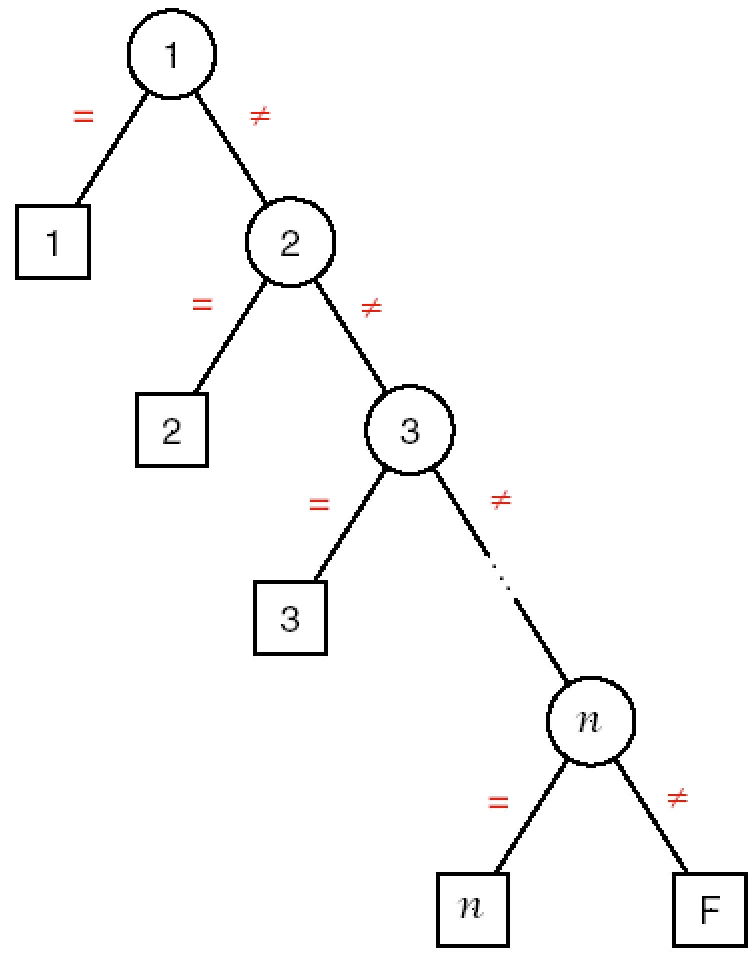
（比较树）

- Definitions
  - The *comparison tree* of an algorithm is obtained by tracing the action of the algorithm, representing each comparison of keys by a *vertex* of the tree (which we draw as a circle). Inside the circle we put the index of the key against which we are comparing the target key.

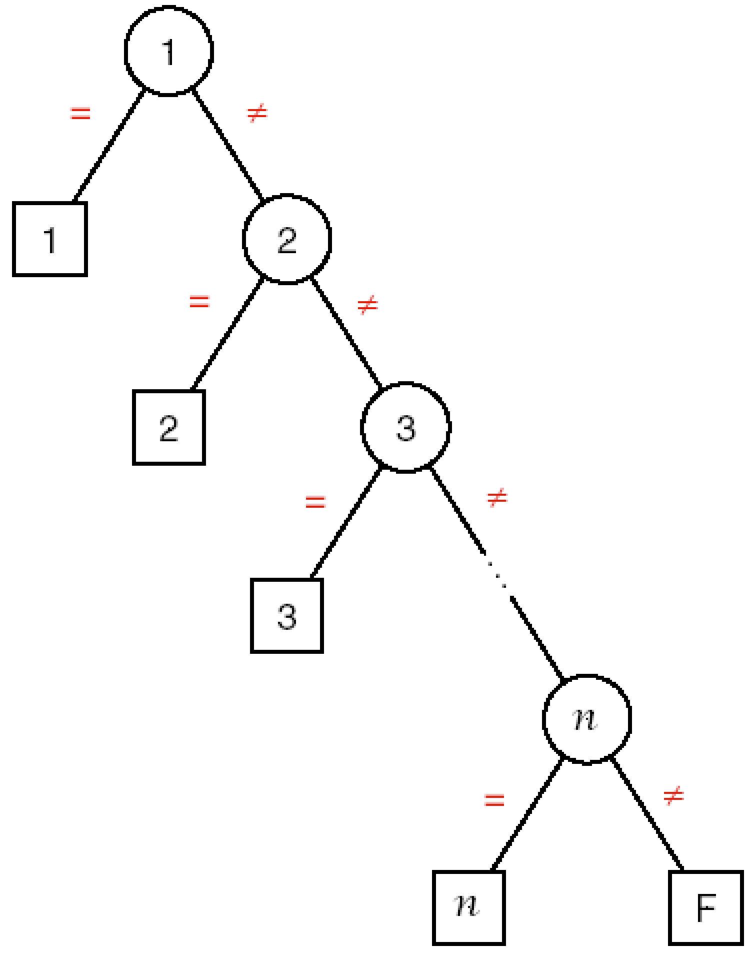
注意在画比较树时，被比较的关键字的序号通常是从1开始编号的。



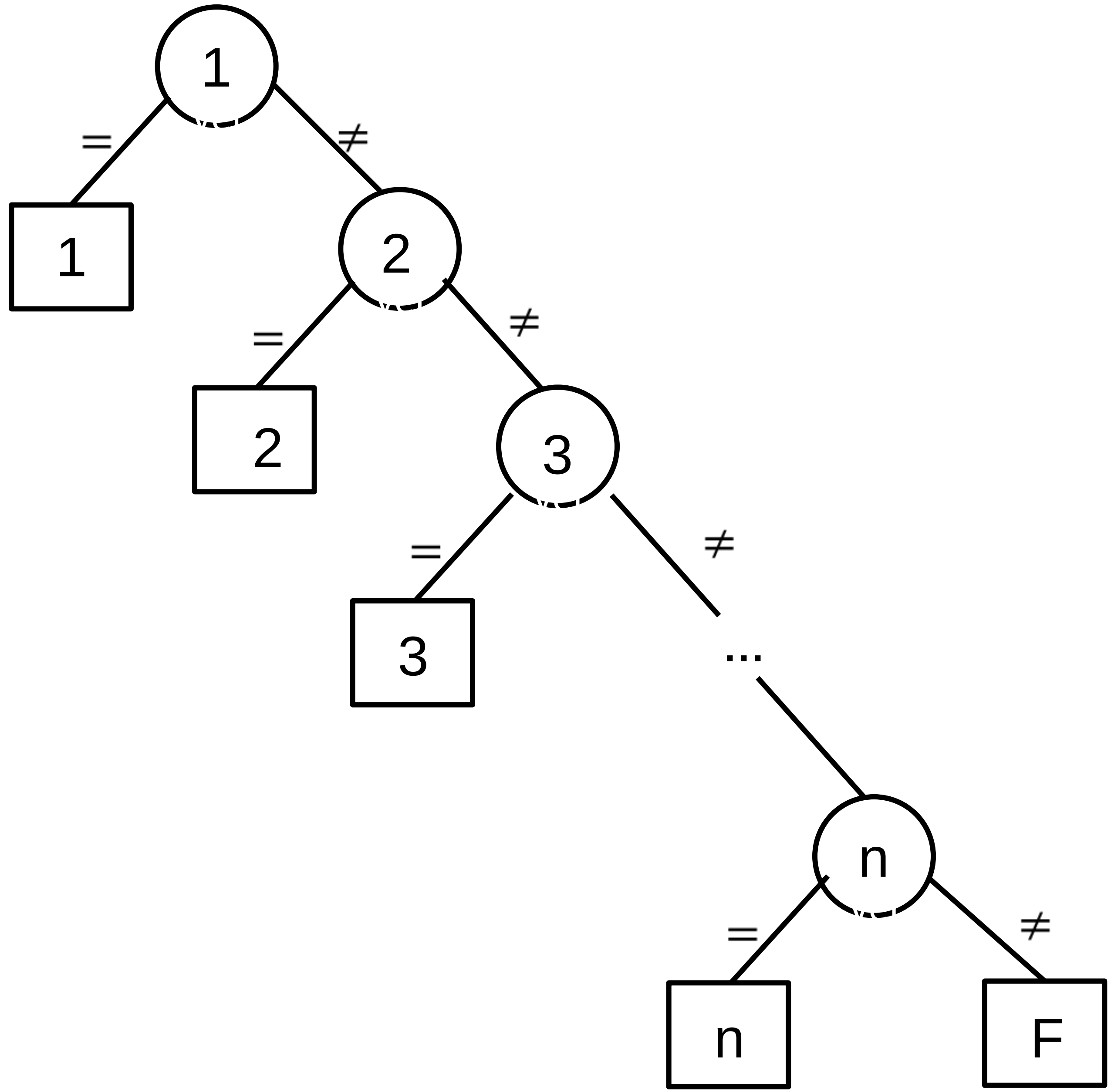
- *Branches (lines)* drawn down from the circle represent the possibl outcomes of the comparison. When the algorithm terminates, put either F (for failu or the location where the target is found at the end of the appropriate branch, which we call a *leaf*, an draw as a square.

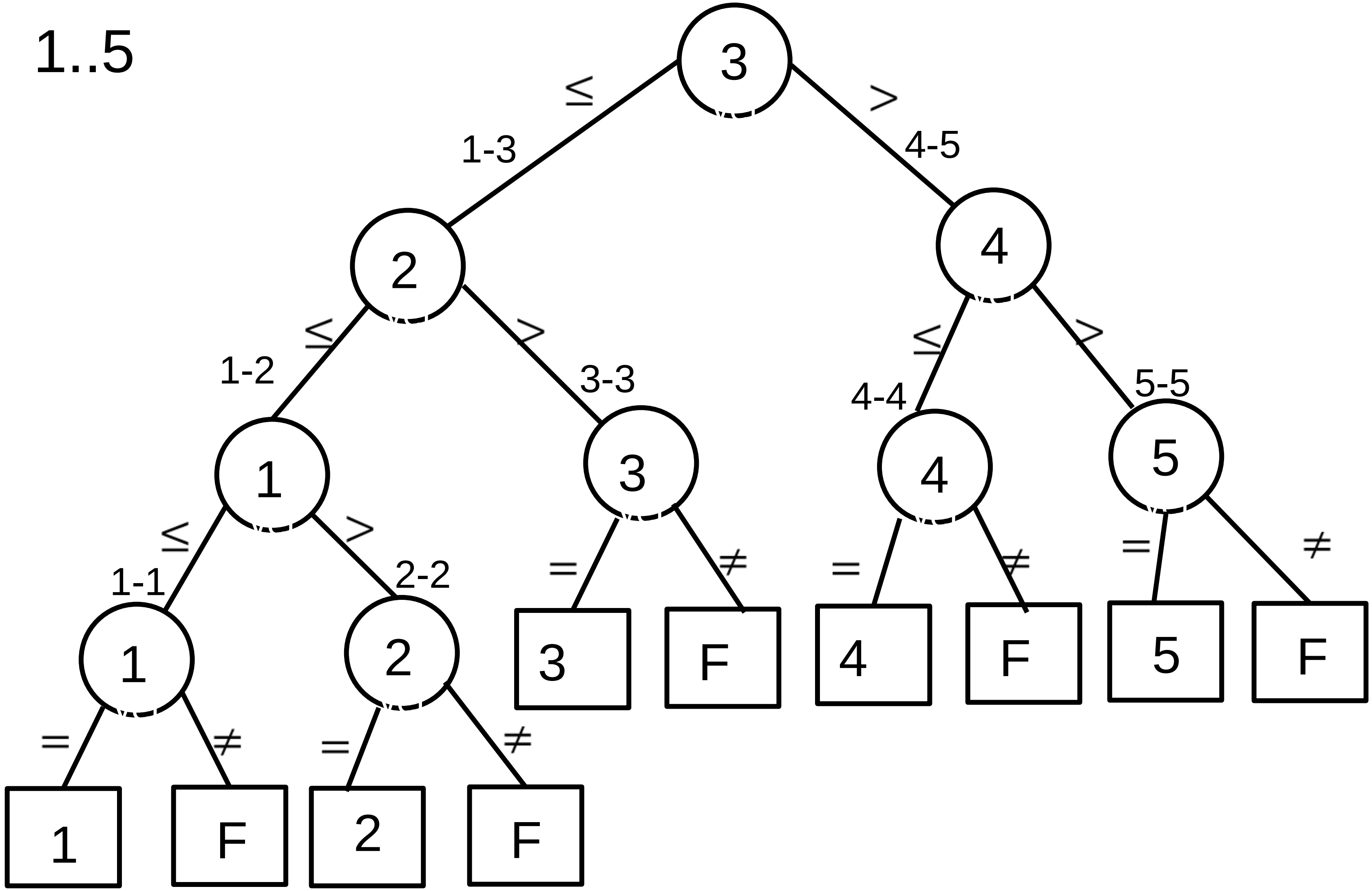


- The remaining vertices are called the *internal vertices* of the tree. The number of comparisons done by an algorithm in a particular search is the number of internal vertices traversed in going from the top of the tree, called its *root*, down the appropriate path to a leaf.

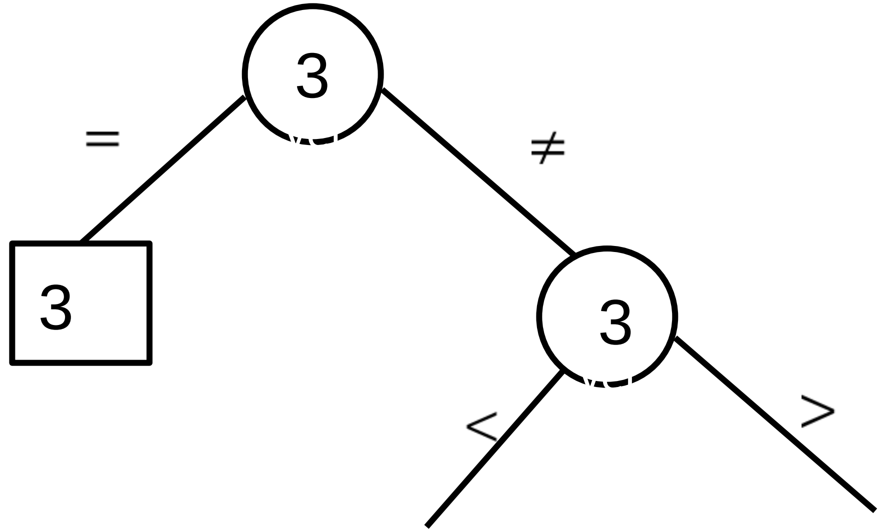


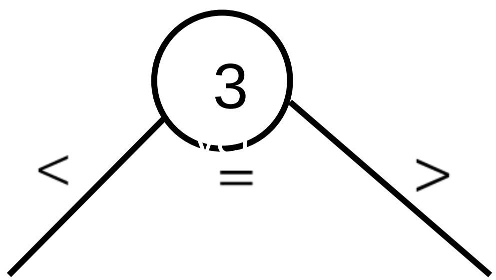
顺序查找比较树画法





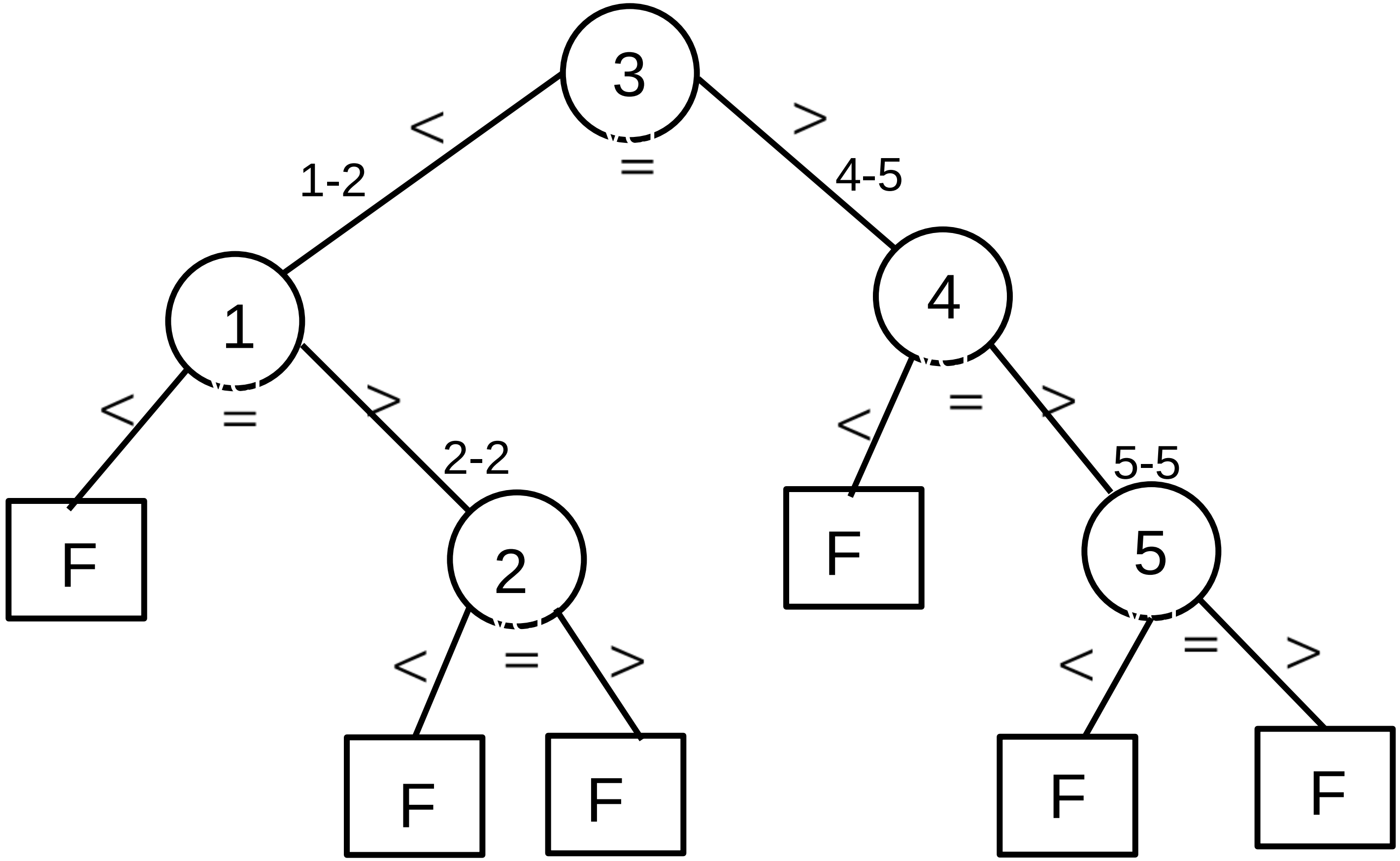
第1种（不识别相等）
二分查找比较树画法





第2种（识别相等）
二分查找比较树画法

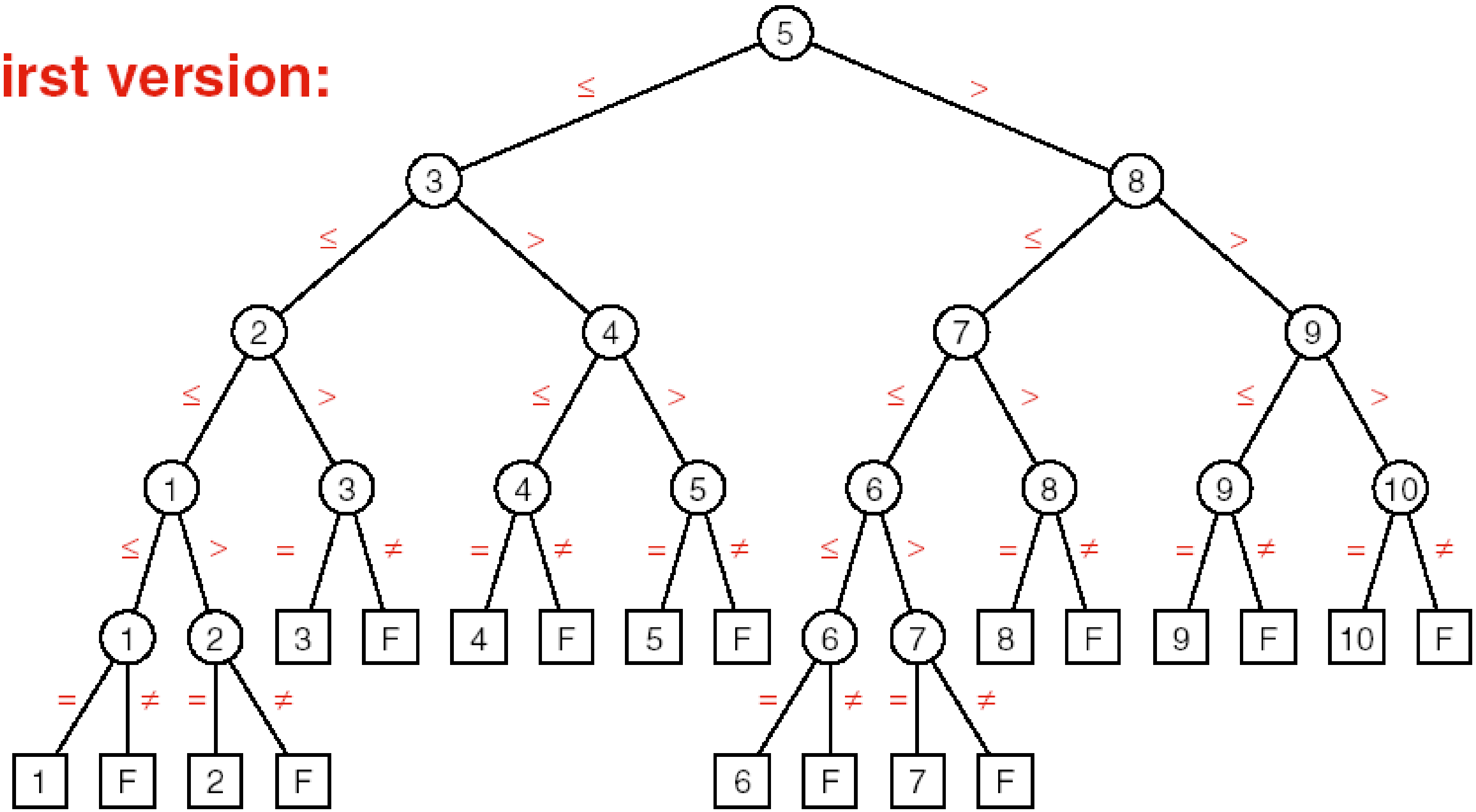
1-5



第2种（识别相等）
二分查找比较树画法

Comparison Trees for Binary Search

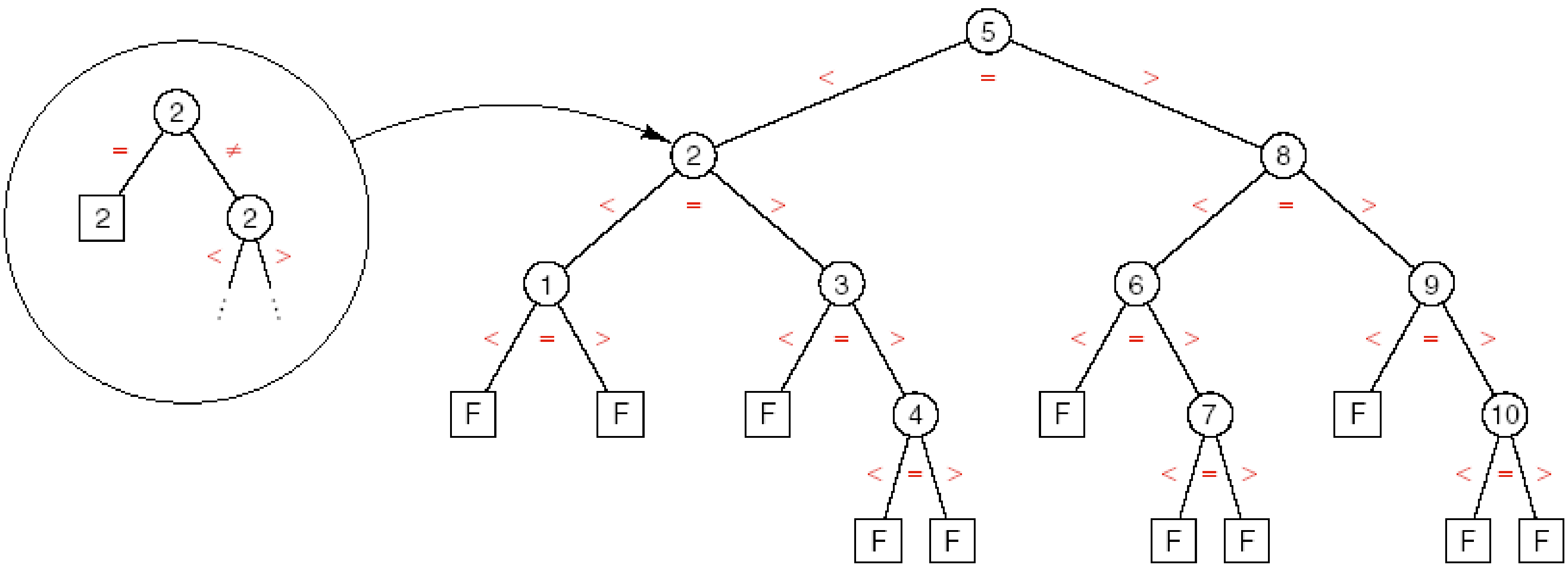
First version:



(成功时，)平均比较次数（5*4+4*6)/10=4.4
(不成功时，)平均比较次数：4.4

Comparison Trees for Binary Search

Second version:



（成功时）平均比较次数：ASL=(1+2*3+4*5+3*7)=4.8
（不成功时）平均比较次数：ASL= (5*6+6*8)/11=78/11

Binary Search Analysis

The number of comparisons of keys done by binary_search_1 in searching a list of n items is approximately

$$\lg n + 1$$

in the worst case and

$$\lg n$$

in the average case. The number of comparisons is essentially independent of whether the search is successful or not.

The number of comparisons done in an unsuccessful search by binary_search_2 is approximately $2 \lg(n + 1)$.

Binary Search Analysis

In a successful search of a list of n entries, binary_search_2 does approximately

$$\frac{2(n+1)}{n}\lg(n+1) - 3$$

comparisons of keys.

The proof of the above results requires the *path length theorem*.

	Successful search	*Unsuccessful search*
binary_search_1	$\lg n + 1$	$\lg n + 1$
binary_search_2	$2\lg n - 3$	$2\lg n$