

그림 1. Insertion Sorting 예제

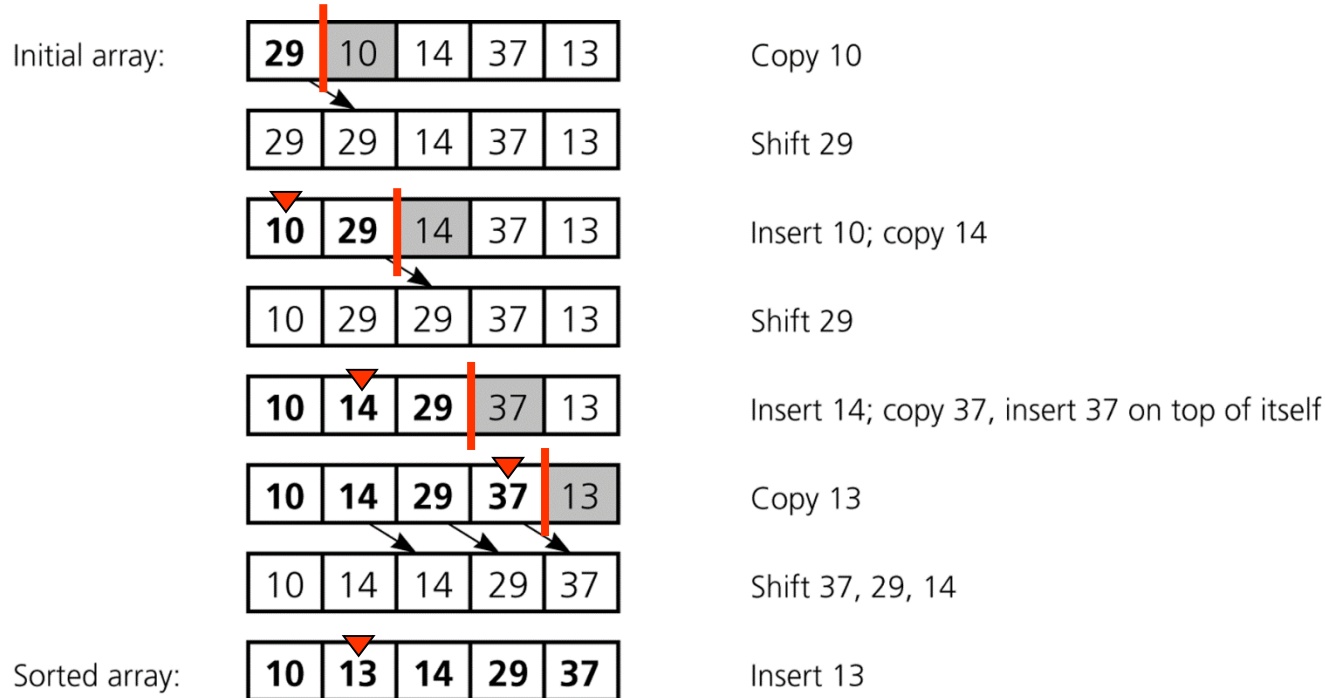
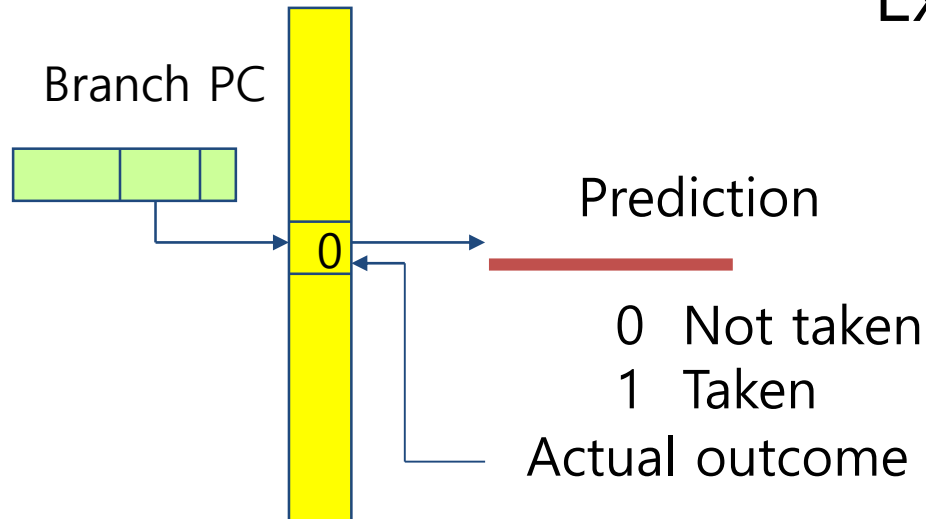
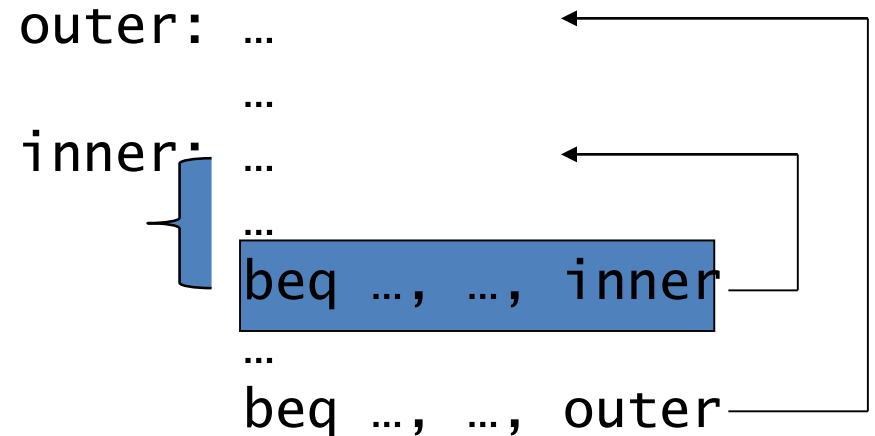


그림 2. Dynamic branch prediction

Branch History Table (BHT)

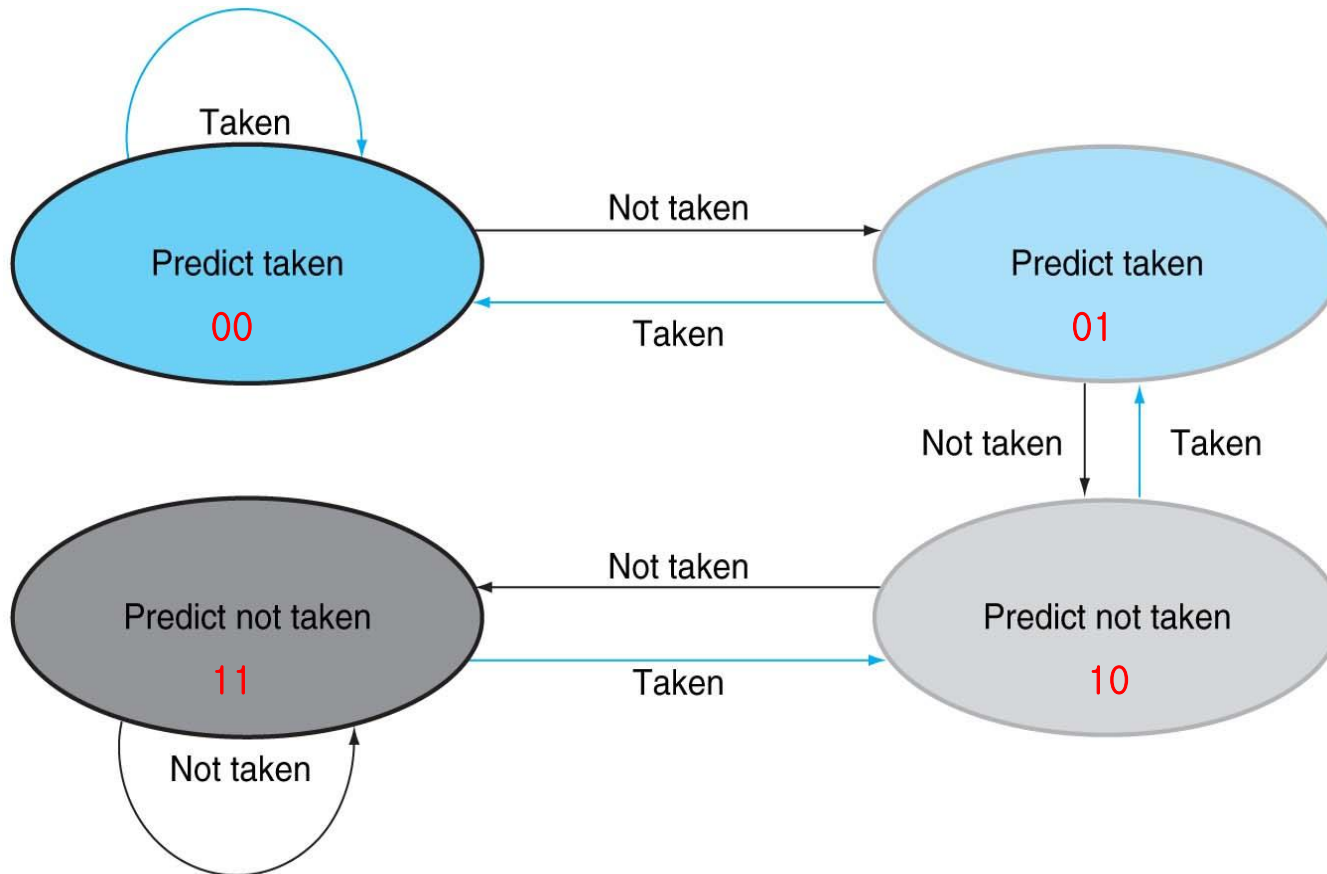


Example: Loops and Prediction



2-bit Branch Prediction Scheme

- Changing prediction only if get misprediction *twice*



보조자료(1/3): Select Algorithm 동작

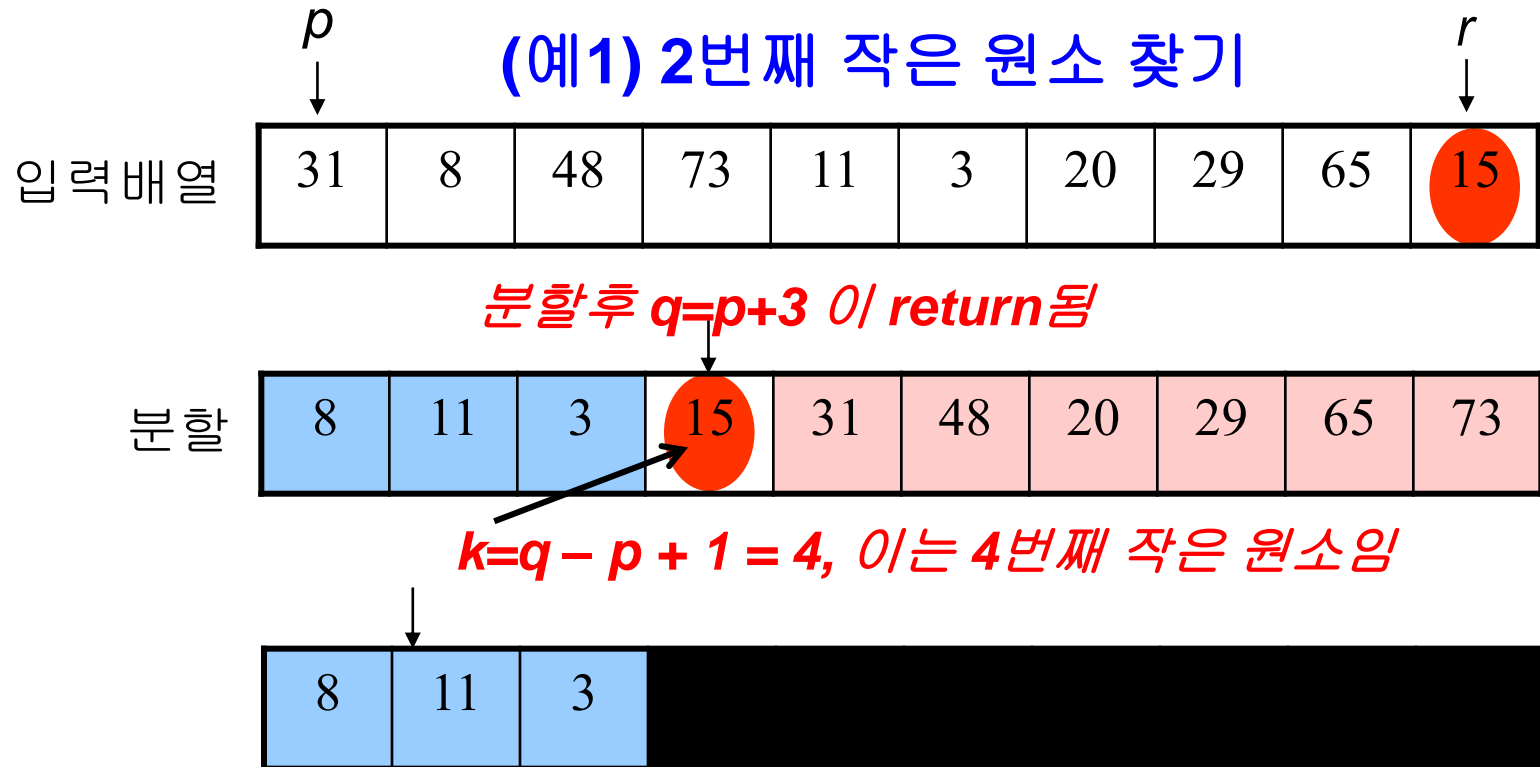
알고리즘 동작원리

1) Quicksort의 partition을 활용

2) Partition후 return 값 "q" = 기준원소의 위치값을 이용하여 해당 partition을 선택한 뒤 recursive call

3) 최종적으로 원소 찾기

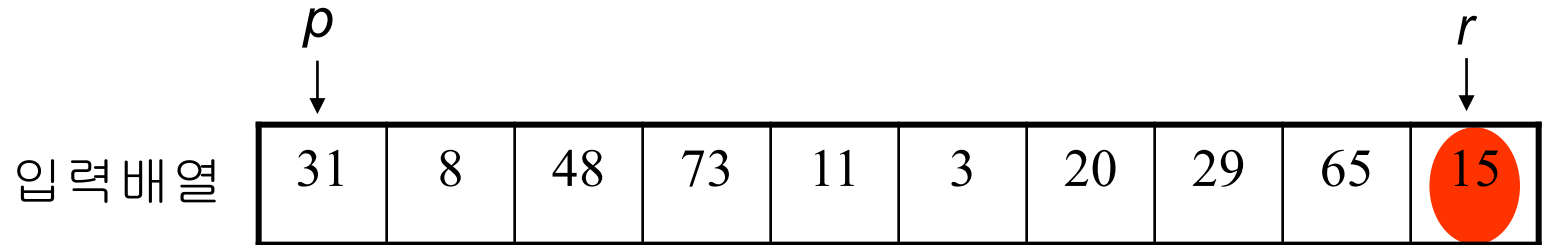
아래 예제에서는 맨 마지막 원소를 pivot로 사용



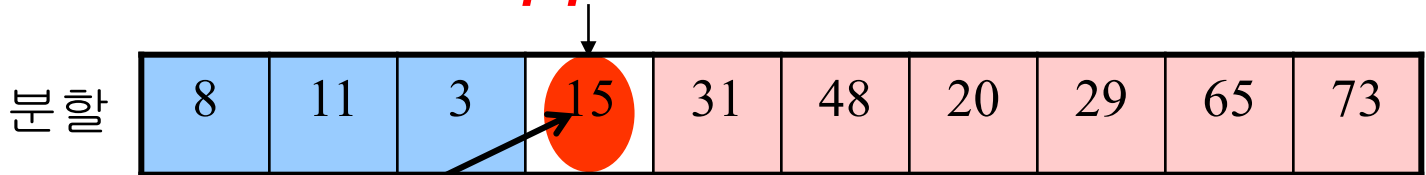
왼쪽 그룹에서 2번째 작은 원소를 찾는다

보조자료(2/3): Select Algorithm 동작

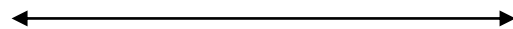
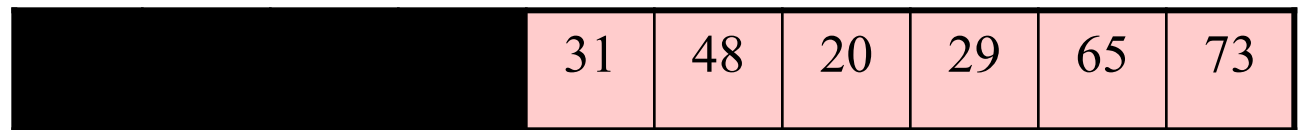
(예2) 7번째 작은 원소 찾기



분할 후 $q = p + 3$ O/ return 됨



$k = q - p + 1 = 4$, 이는 4번째 작은 원소임



4개

오른쪽 그룹에서 3번째 작은 원소를 찾는다

보조자료(3/3): quick sort

```
quicksort( array, first, last ){  
partition( array, first, last );  
quicksort( array, first, j-1 );  
quicksort( array, j+1, last);  
}
```



```
void quicksort(int array[8],int first,int last){  
    int i, j, pivot, temp;
```

```
    if(first<last){  
        pivot=first;  
        i=first;  
        j=last;
```

Partition

```
        while(i<j){  
            while(array[i] <= array[pivot]&& i<last)  
                i++;  
            while(array[j]>array[pivot])  
                j--;  
            if(i<j){  
                temp=array[i];  
                array[i]=array[j];  
                array[j]=temp;  
            }  
        }  
    }
```

```
    temp=array[pivot];  
    array[pivot]=array[j];  
    array[j]=temp;  
    quicksort(array,first,j-1);  
    quicksort(array,j+1,last);
```

```
    }  
}
```