기수 정렬 (Radix sort)

Comparison Sort (비교 정렬)

> N개의 원소의 배열을 정렬하는 가짓수는 N!
> 따라서, Comparison Sort를 통해 생기는 트리의 말단 노드가 N!이상의 노드 갯수를 갖기 위해선 2^h ≥ N!를 만족하는 h(트리의 높이 == 비교정렬의 시간 복잡도)를 가져야 하고, 이 식을 h > O(n logn)을 가져야한다.
>
> O(n logn)을 줄일 수 있는 방법은 Comparison(비교)를 하지 않는 것.

Radix sort

- 데이터를 구성하는 기본 요소(Radix)를 이용한 정렬 방식

> 입력 데이터의 최대값에 따라서 Counting Sort의 비효율성을 개선하기 위해
> Radix Sort를 사용할 수 있음
> 자릿수의 값 별로 (예) 둘째자리, 첫째자리) 정렬을 하므로, 나올 수 있는 값의 최대 사이즈는 9임
> (범위 : 0~9) ⇒ 10진수 표기법 기준

- 시간 복잡도 : O(d * (n+b))
  - d : 정렬할 숫자중 가장 큰 자릿수
  - b : 10 (k와 같으나 10으로 고정됨)
  - Counting Sort의 경우 : O(n+k)로 배열의 최댓값 k에 영향을 받음
- 장점 : 문자열, 정수 정렬 가능
- 단점
  - 자릿수가 없는 것은 정렬할 수 없음. ⇒ 부동소숫점

- 중간 결과를 저장할 bucket 공간이 필요

```
void countSort(int arr[], int n, int exp) {
    int buffer[n];
    int i, count[10] = {0};

    // exp의 자릿수에 해당하는 count 증가
    for (i=0; i<n; i++){
     count[(arr[i]/exp) % 10]++;
    }

    // 누적합 구하기
  for (i = 1; i < 10; i++) {
      count[i] += count[i - 1];
  }
  // 일반적인 Counting sort 과정
  for (i = n - 1; i >= 0; i--) {
      buffer[count[(arr[i]/exp) % 10] - 1] = arr[i];
      count[(arr[i] / exp) % 10]--;
  }
  for (i = 0; i < n; i++){
      arr[i] = buffer[i];
  }
}

int getMAX(int arr[], int n){
    int ret = 0;
    for(int i=0; i<n; i++){
        int cnt = 0;
        int now = arr[i];

        while(now/10){
            now = now / 10;
            cnt++;
        }
        cnt++;
        ret = max(ret, cnt);
    }
```

```
}

void radixsort(int arr[], int n) {
    // 최댓값 자리만큼 돌기
    int m = getMax(arr, n); // 최대값의 자릿수 반환

    // 최댓값을 나눴을 때, 0이 나오면 모든 숫자가 exp의 아래
    for (int exp=1; (m/exp)>0; exp *= 10){
        countSort(arr, n, exp);
    }
}

int main() {
    int arr[8] = { 170, 45, 75, 90, 802, 24, 2, 66};
    int n = sizeof(arr) / sizeof(arr[0]); // 8
    radixsort(arr, n);

    for (int i=0; i<n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}
```

1. 170, 45, 75, 90, 802, 24, 2, 66

안정적(stable)이며, counting sort와 유사

1. 1의 자리 비교

   170, 90, 802, 2, 24, 45, 75, 66

2. 10의 자리 비교

   802, 2, 24, 45, 66, 170, 75, 90

3. 100의 자리 비교

   2, 24, 45, 66, 75, 90, 170, 802