

## 셸 정렬(Shell Sort)

sort

### 정의

- 셸 정렬은 단순 삽입 정렬의 장점을 살리고 단점을 보완하여 좀 더 빠르게 정렬하는 알고리즘이다.
- 일정한 간격으로 서로 떨어져 있는 두 요소를 그룹으로 묶어 대략적으로 정렬하고 간격을 좁혀 그룹의 수를 줄이면서 정렬을 반복하여 이동 횟수를 줄이는게 특징이다.
- 불안정 정렬의 종류 중 하나이다.
- 주의 할 점으로 효과적인 정렬을 하기 위해 증분 값  $h$ 가 서로 배수가 되지 않아야 효과적으로 정렬 할 수 있다.(  $h = \dots 123 \rightarrow 40 \rightarrow 13 \rightarrow 4 \rightarrow 1$  )
- 위 수열을 뒤에서부터 살펴보면 1부터 시작해 3배한 값에 1을 더하는 수열이라는것을 알 수 있다.

### 시간 복잡도

- 기존 단순 삽입 정렬보다 훨씬 빠른  $O(n^{1.25})$ 의 시간 복잡도를 가진다.

```

class ShellSort2 {
    //--- 셸 정렬 ---//
    static void shellSort(int[] a, int n) {
        int h;
        // --- 1부터 시작해 값을 3배하고 1을 더하면서 n을 넘지 않는 가장 큰 값 --- //
        for (h = 1; h < n; h = h * 3 + 1)
            ;

        // --- 반복할 때마다 h값을 3으로 나눠 마지막 h값은 1이된다.
        for ( ; h > 0; h /= 3)
            for (int i = h; i < n; i++) {
                int j;
                int tmp = a[i];
                // --- 정렬 조건 : 반복문 종료 조건
                for (j = i - h; j ≥ 0 && a[j] > tmp; j -= h)
                    a[j + h] = a[j];
                a[j + h] = tmp;
            }
    }

    public static void main(String[] args) {
        Scanner stdin = new Scanner(System.in);

        System.out.println("셸 정렬(버전 2)");
        System.out.print("요솜수: ");
        int nx = stdin.nextInt();
        int[] x = new int[nx];

        for (int i = 0; i < nx; i++) {
            System.out.print("x[" + i + "]: ");
            x[i] = stdin.nextInt();
        }

        shellSort(x, nx);                // 배열 x를 셸정렬

        System.out.println("오름차순으로 정렬했습니다.");
        for (int i = 0; i < nx; i++)
            System.out.println("x[" + i + "]=" + x[i]);
    }
}

```