문제해결기법

Week-01 Practice

컴퓨팅 단위

- 이진수, 십진수 대응
  - K → $2^{10} = 1024 \approx 10^3$ (일천)
  - M → $2^{20} \approx 10^6$ (백만)
  - G → $2^{30} \approx 10^9$ (십억)

연산자 +

```
public class Test {
	public static void main(String[] args) {
		String	s1="123"+"456";
		System.out.println(s1);

		String	s2="123"+456;
		System.out.println(s2);

		String	s3=123+"456";
		System.out.println(s3);

		String	s4=""+123+456;
		System.out.println(s4);

		String	s5=123+456+"";
		System.out.println(s5);
	}
}
```

정수 자료형

- 정수 자료형
  - 3비트 2의 보수 정수 표현
    - -4, -3, -2, -1, 0, 1, 2, 3

$$-4 = -2^2$$

$$3 = 2^2 - 1$$

이진수	십진수
000	0
001	1
010	2
011	3
100	-4
101	-3
110	-2
111	-1

  - 32비트 2의 보수 정수 표현
    - 최소값: -2^{31}
    - 최대값: $2^{31}-1$

자바 정수(int) 값 범위

```
public class Test {
    public static void main(String[] args) {
        System.out.println(Integer.MAX_VALUE);
        System.out.println(Integer.MIN_VALUE);
    }
}
```

```
2147483647
-2147483648
```

자바 정수(long) 값 범위


```
public class Test {
    public static void main(String[] args) {
        System.out.println(Long.MAX_VALUE);
        System.out.println(Long.MIN_VALUE);
    }
}
```

```
9223372036854775807
-9223372036854775808
```

자바 BigInteger

- long 자료형 범위 이상의 정수 표현

```
public class Test {
	public static void main(String[] args) {
		String	s1="99999999999999999999999999999999999999999999888888888888888888";
		String	s2="88888888888888888888888888888888888877777777777777777";
		BigInteger n1=new BigInteger(s1);
		BigInteger n2=new BigInteger(s2);
		System.out.println(n1.add(n2));
	}
}
```

자바 배열 생성

```
public class Test {
	public static void main(String[] args) {
		int	n[];
		n=new int[3];
		n[0]=77;
		n[1]=88;
		n[2]=99;
		for (int i = 0; i < n.length; i++) {
			System.out.println(n[i]);
		}
	}
}
```

```
public class Test {
	public static void main(String[] args) {
		int	n[]=new int[3];
		n[0]=77;
		n[1]=88;
		n[2]=99;
		for (int v : n) {
			System.out.println(v);
		}
	}
}
```

```
public class Test {
	public static void main(String[] args) {
		int	n[]={77,88,99};
		System.out.println(Arrays.toString(n));
	}
}
```

최대값 탐색

```
public class Test {
	public static void main(String[] args) {
		int   n[]= {32,54,65,21,10};
		int   max=n[0];
		for (int i = 0; i < n.length; i++) {
			if(n[i]>max) max=n[i];
		}
		System.out.println(max);
	}
}
```

총 비교 횟수(n=5): 5
총 비교 횟수(n=1000): 1000

$O(n)$

0	1	2	3	4	max
32	54	65	21	10	**32**
max < 32 ?					**32**
	max < 54 ? max를 54로 변경				**54**
		max < 65 ? max를 65로 변경			**65**
			max < 21 ?		**65**
				max < 10 ?	**65**

버블 정렬

Reference: https://en.wikipedia.org/wiki/Bubble_sort

					오름차순 정렬 가정
4	3	5	2	1	최초 상태
4	3	5	2	1	4,3 비교 후 교환
3	4	5	2	1	4,5 비교
3	4	5	2	1	5,2 비교 후 교환
3	4	2	5	1	5,1 비교 후 교환
3	4	2	1	**5**	n=5인 경우, 총 4번 비교 후 최대값(5) 위치 결정
3	2	1	**4**	**5**	총 3회 비교 후 두번째 큰 값(4) 위치 결정
2	1	**3**	**4**	**5**	총 2회 비교 후 세번째 큰 값(3) 위치 결정
1	**2**	**3**	**4**	**5**	총 1회 비교 후 네번째 큰 값(2) 위치 결정

총 비교 횟수(n=5): 4+3+2+1
총 비교 횟수(n=1000): 999+998+ … +2+1

$$1 + 2 + \cdots (n-1) = \frac{n(n-1)}{2} = O(n^2)$$

정렬 알고리즘과 시간복잡도

Reference: https://en.wikipedia.org/wiki/Sorting_algorithm

- 정렬 알고리즘과 시간복잡도

최악의 경우 시간복잡도	정렬 알고리즘
$O(n^2)$	버블정렬, 선택정렬, 삽입정렬, 퀵정렬
$O(n \log n)$	합병정렬, 힙정렬

References

- C로 쓴 자료구조론 (Fundamentals of Data Structures in C, Horowitz et al.). 이석호 역. 사이텍미디어. 1993.
- 쉽게 배우는 알고리즘: 관계 중심의 사고법. 문병로. 한빛아카데미. 2013.
- C언어로 쉽게 풀어 쓴 자료구조. 천인국 외 2인. 생능출판사. 2017.