자료구조 <Quiz 1>

2014027108 신동훈

1.

처음 f(101) 을 호출하면, 에서드 f 는 다음과 같이 재귀적으로 호출된다.

f(101) → f(50) → f(25) → f(12) → f(6) → f(3) → f(1)

따라서 총 7번 호출되며, f(1)일 때 1이 return 된 후, 호출된 수 만큼 1이 더해져 return 되게 때문에 답은 7이 된다.

답: 7

2.

n이 1일때는 if문 속 비교연산 (==) 1회와, return 1회, 총 2의 시간이 필요하며

n이 1보다 큰 경우 if문 속 비교연산(==) 1회, 나누기 산술연산(/) 1회, 함수의 call 1회, 더하기 산술연산 (+) 1회, return 1회와, 함수의 call로 인해 발생되는 연산의 수인 T(n/2)회, 즉 총 T(n/2)+5 의 시간이 필요하므로 점화식 T(n)은 다음과 같다.

$$T(n) = \begin{cases} 2 & \text{if } n=1 \\ T(n/2)+5 & \text{if } n>1 \end{cases}$$

3. n이 2의 승수, 즉 $n=2^k$ ($k \geq 0$)의 형태를 가질 때 T(n)은 다음과 같다.

$$T(n) = T(2^k) = T(2^k/2)+5 = T(2^{k-1})+5$$
$$= (T(2^{k-1}/2)+5)+5 = T(2^{k-2})+5\cdot 2$$
$$= (T(2^{k-2}/2)+5)+5\cdot 2 = T(2^{k-3})+5\cdot 3$$
$$\vdots$$
$$= T(2^0)+5\cdot k = 2+5\cdot k$$

즉 $T(n) = T(2^k) = 2+5\cdot k$

그런데, $n=2^k$ 이므로 $k = \log_2(n)$ 이다.

그러므로 $T(n) = T(2^k) = 5\cdot k+2 = 5\cdot \log_2(n)+2$

답: $T(n) = 5\cdot \log_2(n)+2$

4. ArrayList는 배열을 사용하여 List를 구현하였기에 이에 대한 ListIterator는 배열의 인덱스를 이용하여 구현할 수 있다.

답!

```
private int _nextPosition; // 다음 위치를 나타내는 인덱스

Private ListIterator() {
    this._nextPosition = 0; // index를 0으로 초기화
}

@Override
public boolean hasNext() {
    return this._nextPosition < ArrayList.this.size(); // 다음 위치의 index가 ArrayList의 size보다 작으면 다음 원소가 존재
}

@Override
public E next() {
    return (this.hasNext()) ? ArrayList.this.elements()[this._nextPosition++] : null;
}
```

5. ListIterator 의 hasNext() 과 next()를 사용하여 다음과 같이 작성할 수 있다.

답:

```
Iterator<Integer> iterator = myList.iterator();

while (iterator.hasNext()) {
    sum += iterator.next();
}
```