

Pernyataan yang benar mengenai ketiga method tersebut adalah ...

- A. zzz1, zzz2, dan zzz3 semuanya bukan tail recursive.
- B. zzz1 adalah rekursif tapi bukan tail recursive.
- C. zzz1 dan zzz2 merupakan tail recursive namun zzz3 bukan.
- D. zzz1, zzz2, dan zzz3 semuanya recursive.

8. Perhatikan method rekursif berikut:

```
void konversi(int number, int base) {
    if (number <= 0)
        return "";
    else
        return konversi(number/base, base) + (number%base);
}
```

Jika number berharga N (suatu variabel) sementara b adalah konstan maka kompleksitas waktu method rekursif tersebut adalah ...

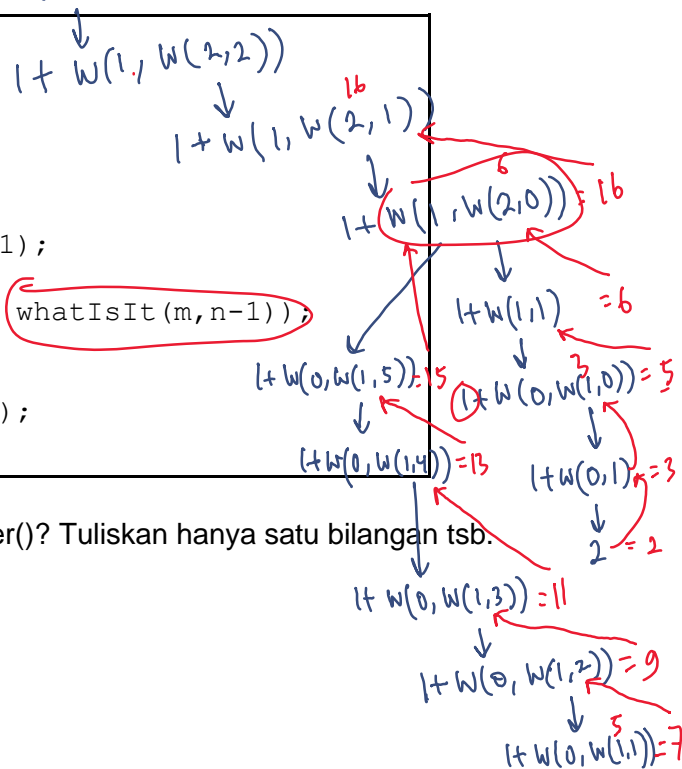
- A. $O(\log N)$
- B. $O(1)$
- C. $O(N)$
- D. $O(N \log N)$

$\log_b N$

9. Perhatikan implementasi fungsi rekursif berikut.

```
int whatIsIt(int m, int n) {
    if (m < 0 || n < 0) return -1;
    if (m == 0)
        return n+1;
    else
        if (n==0)
            return 1+ whatIsIt (m-1, 1);
        else
            return 1+ whatIsIt (m-1, whatIsIt(m, n-1));
}
void caller() {
    System.out.println(whatIsIt(2, 3));
}
```

$w(2, 3)$



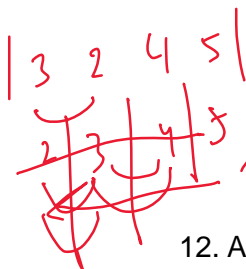
Bilangan apa yang akan dicetak oleh method caller()? Tuliskan hanya satu bilangan tsb.

10. Diberikan suatu method rekursif sebagai berikut.

```
void half(int [] numbers, int left, int right) {  
    if (left >= right) {  
        System.out.println(numbers[right]);  
    }  
    else {  
        int mid = (left+right)/2;  
        half (numbers, left, mid);  
        half (numbers, mid+1, right);  
    }  
}
```

Periksalah pernyataan-pernyataan berikut mengenai method tersebut (jika panjang array numbers adalah N dan method awalnya dipanggil dengan `half(numbers, 0, N-1)` dan tunjukkan mana yang benar.

- ☒ A. Algoritma method tersebut adalah $O(N \log N)$ karena mirip merge sort.
 - ☐ B. Time complexity algoritma method ini memiliki best case $O(\log N)$ dan worst-case $O(N \log N)$.
 - ☐ C. Algoritma itu akan me-return harga median (data yang ada di tengah array).
 - ☐ D. Time complexity algoritma method ini adalah linear atau $O(N)$ karena pencetakan di baris ke 4 dilakukan sebanyak N.
11. Algoritma bubble sort versi dasarnya melakukan penukaran pada pasangan-pasangan dengan urutan dari kiri ke kanan. Seandainya algoritma bubble-sort dimodifikasi sehingga untuk setiap iterasi (outer loop) penukaran dilakukan pada kedua arah secara bergantian (1 inner loop untuk penukaran dari kiri ke kanan, kemudian 1 inner loop untuk penukaran dari kanan ke kiri), kondisi apakah yang akan terjadi?



- ☒ A. Kompleksitas waktu eksekusi tetap $O(N^2)$ dan waktu eksekusi tidak berubah secara signifikan.
- ☐ B. Waktu eksekusi akan bertambah menjadi duakali semula tapi tetap $O(N^2)$.
- ☒ C. Waktu eksekusi akan dua kali menjadi lebih lama walaupun tetap $O(N^2)$.
- ☐ D. Kompleksitas akan berubah menjadi logaritmis.

$2N^2$
 $O(N^2)$

12. Algoritma merge sort memerlukan memory space tambahan. **Pernyataan yang salah mengenai memory space** tersebut adalah:

- ☐ A. Memory space diperlukan untuk hasil operasi merging dua bagian yang sudah terurut namun terpisah sebelumnya.
- ☐ B. Ukuran total memory space tersebut adalah dua kali ukuran array yang sebenarnya.
- ☒ C. Memory space tambahan digunakan untuk menampung hasil proses algoritma **partisi** dari merge sort.
- ☐ D. Kompleksitas waktu dalam melakukan operasi merging adalah $O(N)$, jika N adalah banyaknya data yang diurutkan.

quick sort

13. Misalnya kita menggunakan Quicksort untuk mengurutkan sebuah array secara ascending. Kemudian, diketahui hasil iterasi pertama dalam algoritma Quicksort menghasilkan array sebagai berikut:

14	31	35	61	62	66	68	69
31	14	35	62	61	66	69	68

Pernyataan yang benar mengenai pivot pada iterasi pertama tersebut adalah:

- A. Data yang mungkin menjadi pivot hanya 35 dan 69.
 - ☒ B. Data yang mungkin menjadi pivot adalah 35 dan 66.
 - C. Data yang mungkin menjadi pivot hanya 66.
 - D. Data yang mungkin menjadi pivot adalah 62 dan 68.
14. List di bawah ini merupakan 2 tahap/iterasi sebuah algoritma sorting. Algoritma apakah yang dimaksud?

Data Awal:	134	112	101	105	132	156
Setelah Iterasi 1:	112	134	101	105	132	156
Setelah Iterasi 2:	101	112	134	105	132	156
Setelah Iterasi 3:	105	101	112	134	132	156

- A. Selection Sort
 - B. Bubble Sort
 - C. Insertion Sort
 - D. Shellsort
15. Suatu algoritma **Quicksort** melakukan partisi dengan algoritma *Partisi-1* (sesuai dijelaskan di materi kuliah) dengan pivot adalah elemen pertama ruas data yang diberikan. Diberikan ruas data awal (sebelum algoritma partisi dijalankan) dan isi ruas data setelah iterasi ketiga (angka yang **bold** adalah pivot):

Ruas Awal:	42	81	35	71	39	41	25	34
Iterasi 1:	34	81	35	71	39	41	25	42
Iterasi 2:	34	42	35	71	39	41	25	81
Iterasi 3:	34	25	35	71	39	41	42	81

Jika iterasi dilanjutkan hingga algoritma partisi itu selesai, tuliskanlah hasil akhir partisi tersebut.

2. Diberikan data berisi 20 bilangan bulat dengan urutan awal sebagai berikut:

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Awal	80	99	87	65	66	58	90	85	64	34	57	89	56	54	25

Simulasikan algoritma Shell-sort pada data berikut ini dengan deret gap = 5,3,1.
Isilah tabel berikut untuk menunjukkan hasil sort setelah Sort-5 Sort-3 dan Sort-1.

Indeks	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Awal	80	99	87	65	66	58	90	85	64	34	57	89	56	54	25
Setelah Sort-5	57	89	80	54	25	58	90	85	64	34	80	99	87	65	66
Setelah Sort-3	34	25	80	54	65	58	57	80	64	87	85	66	90	89	99
Setelah Sort-1	25	34	54	56	57	58	64	65	66	80	85	87	89	90	99