



Clara Aurelia Setiady - 2306217304 Clara ▾



Home > My courses > PROG. S1 FAK. REGULER > REG - Gasal 2024/2025 > [Reg] Struktur Data & Algoritma (A,B,C,D,E,F) ... > Pekan 3: Rekursif > CP03 Rekursif

Started on Thursday, 12 September 2024, 11:19 PM

State Finished

Completed on Saturday, 14 September 2024, 11:59 PM

Time taken 2 days

Grade 8.00 out of 10.00 (80%)

Question 1

Correct

Mark 1.00 out of 1.00

Manakah dari berikut ini yang merupakan masalah umum yang dihadapi dalam pemrograman rekursif?

- ☐ a. Rekursi tidak dapat digunakan untuk masalah yang melibatkan struktur data seperti tree atau graf.
- ☐ b. . Rekursi selalu lebih efisien dibandingkan dengan pendekatan iteratif.
- ☐ c. Rekursi sering kali menghasilkan kode yang lebih panjang dibandingkan dengan iterasi.
- ☒ d. Rekursi dapat menyebabkan stack overflow jika pemanggilan rekursif terlalu dalam. ✓

Your answer is correct.

The correct answer is: Rekursi dapat menyebabkan stack overflow jika pemanggilan rekursif terlalu dalam.

Question 2

Incorrect

Mark 0.00 out of 1.00

Perhatikan potongan kode program berikut ini:

```
public int recursion(int n, int m) {  
    if (n <= 0) {  
        return m;  
    } else if (m <= 0) {  
        return n;  
    } else {  
        return 1 + recursion(n - 1, m) + recursion(n, m - 1) - recursion(n - 1, m - 1);  
    }  
}
```

Apa kompleksitas waktu dari fungsi `recursion` di atas?

- ☐ a. $O(2^n)$
- ☐ b. $O(n+m)$
- ☐ c. $O(nm)$
- ☒ d. $O(2^{(m+n)})$ ❌

Your answer is incorrect.

The correct answer is: $O(n+m)$

Question 3

Correct

Mark 1.00 out of 1.00

Ciri masalah yang dapat diselesaikan secara rekursif adalah masalah itu dapat di-reduksi menjadi satu atau lebih masalah-masalah serupa yang lebih kecil. Manakah dari berikut ini yang **bukan** contoh dari ciri tersebut?

- ☐ a. Setiap sub-masalah dapat dipecah lagi hingga mencapai kondisi dasar (base-case).
- ☒ b. Masalah memerlukan pendekatan iteratif untuk penyelesaiannya. ✓
- ☐ c. Masalah dapat dipecah menjadi sub-masalah yang lebih kecil dan serupa.
- ☐ d. Penyelesaian sub-masalah dapat digabungkan untuk menyelesaikan masalah asli.

Your answer is correct.

The correct answer is: Masalah memerlukan pendekatan iteratif untuk penyelesaiannya.

Question 4

Incorrect

Mark 0.00 out of 1.00

Perhatikan potongan kode program berikut ini:

```
public int recMethod(int[] array, int idx1, int idx2) {  
    if (idx1 >= idx2) {  
        return array[idx1];  
    }  
    int v1 = (idx1 + idx2) / 2;  
    int v2 = recMethod(array, idx1, v1);  
    int v3 = recMethod(array, v1 + 1, idx2);  
    return Math.max(v2, v3);  
}
```

Berapa kompleksitas dari fungsi di atas?

- ☐ a. $O(n \log n)$
- ☐ b. $O(n^2)$
- ☐ c. $O(\log n)$
- ☒ d. $O(n)$ ❌

Your answer is incorrect.

The correct answer is: $O(n \log n)$

Question 5

Correct

Mark 1.00 out of 1.00

Perhatikan potongan-potongan fungsi rekursif berikut ini:

```
int funcA(int x) {  
    if (x <= 1) {  
        return 1;  
    } else {  
        return x * funcA(x - 1);  
    }  
}
```

```
int funcB(int y) {  
    if (y <= 1) {  
        return y;  
    } else {  
        return funcB(y - 1) + funcB(y - 2);  
    }  
}
```

```
int funcC(int arr[], int left, int right) {  
    if (left >= right) {  
        return 0;  
    } else {  
        int mid = (left + right) / 2;  
        return funcC(arr, left, mid) + funcC(arr, mid + 1, right);  
    }  
}
```

```
int funcD(int n) {  
    if (n <= 0) {  
        return 0;  
    } else {  
        return n + funcD(n - 1);  
    }  
}
```

Manakah dari fungsi-fungsi di atas yang menggunakan pendekatan divide and conquer?

- ☐ a. funcD
- ☐ b. funcB
- ☒ c. funcC ✓
- ☐ d. funcA

Your answer is correct.

The correct answer is: funcC

Question 6

Correct

Mark 1.00 out of 1.00



Bagaimana dynamic programming meningkatkan efisiensi dari algoritma rekursif?

- ☒ a. Dengan menyimpan hasil dari sub-masalah yang telah dihitung untuk menghindari komputasi berulang.
- ☐ b. Dengan menghindari penggunaan rekursi dan hanya menggunakan iterasi.
- ☐ c. Dengan mengurangi jumlah base-case yang diperlukan dalam rekursi.
- ☐ d. Dengan menggunakan lebih banyak memori untuk menyimpan variabel lokal.



Your answer is correct.

The correct answer is: Dengan menyimpan hasil dari sub-masalah yang telah dihitung untuk menghindari komputasi berulang.

Question 7

Correct

Mark 1.00 out of 1.00

Perhatikan potongan kode Java berikut ini:

```
public class ExampleClass {  
    public static void main(String[] args) {  
        System.out.println(exampleMethod(5));  
    }  
  
    public static int exampleMethod(int n) {  
        if (n <= 0) {  
            return 0;  
        } else {  
            return n + exampleMethod(n - 1) + exampleMethod( n );  
        }  
    }  
}
```

Pernyataan yang benar mengenai kode di atas adalah ...

- ☐ a. Kode akan menghasilkan kesalahan kompilasi karena kesalahan sintaks.
- ☒ b. Kode akan menyebabkan stack overflow karena infinite recursion. ✓
- ☐ c. Kode akan berhenti setelah beberapa iterasi karena kondisi dasar yang benar.
- ☐ d. Kode akan menghasilkan hasil yang benar tanpa masalah.

Your answer is correct.

The correct answer is: Kode akan menyebabkan stack overflow karena infinite recursion.

Question 8

Correct

Mark 1.00 out of 1.00

Perhatikan potongan kode program berikut ini:

```
public int exampleFunction(int n) {  
    if (n <= 1) {  
        return 1;  
    } else if (n % 2 == 0) {  
        return exampleFunction (n / 2) + exampleFunction (n / 2 - 1);  
    } else {  
        return exampleFunction (n - 1) + exampleFunction (n - 3);  
    }  
}
```

Manakah dari berikut ini yang merupakan kompleksitas waktu dari fungsi exampleFunction?

- ☐ a. $O(\log_2 n)$
- ☐ b. $O(n^2)$
- ☒ c. $O(2^n)$ ✓
- ☐ d. $O(n)$

Your answer is correct.

The correct answer is: $O(2^n)$

Question 9

Correct

Mark 1.00 out of 1.00

Manakah dari berikut ini yang merupakan hal penting yang harus diperhatikan dalam pembuatan fungsi rekursif?

- ☒ a. Menentukan kondisi dasar (base case) untuk menghentikan rekursi. ✓
- ☐ b. Menggunakan loop iteratif di dalam fungsi rekursif untuk efisiensi.
- ☐ c. Memastikan fungsi rekursif hanya memanggil dirinya sendiri sekali.
- ☐ d. Menghindari penggunaan variabel global dalam fungsi rekursif.

Your answer is correct.

The correct answer is: Menentukan kondisi dasar (base case) untuk menghentikan rekursi.

Question 10

Correct

Mark 1.00 out of 1.00

Manakah dari berikut ini yang merupakan karakteristik dari fungsi tail recursive?

- ☒ a. Fungsi tail recursive melakukan pemanggilan rekursif sebagai operasi terakhir sebelum kembali ke pemanggil. ✓
- ☐ b. . Fungsi tail recursive selalu memiliki lebih dari satu base-case.
- ☐ c. Fungsi tail recursive tidak dapat diubah menjadi fungsi iteratif.
- ☐ d.

Fungsi tail recursive selalu lebih lambat dibandingkan dengan fungsi rekursif biasa.

Your answer is correct.

The correct answer is: Fungsi tail recursive melakukan pemanggilan rekursif sebagai operasi terakhir sebelum kembali ke pemanggil.