

# Recursive, Number Theory, Sorting & Searching

m.rezahidayat.rh@gmail.com [Switch account](#)

 Draft saved

\* Required

Email \*

m.rezahidayat.rh@gmail.com

dari fungsi cek bilangan prima, apa yang menyebabkan fungsi berikut jauh lebih efisien dibanding \* algoritma pengecekan bilangan prima umumnya..

```
func checkPrime(number int) bool {
    if number < 2 {
        return false
    }
    sqrtNumber := int(math.Sqrt(float64(number)))
    for i := 2; i <= sqrtNumber; i++ {
        if number % i == 0 {
            return false
        }
    }
    return true
}
```

- ☐ number % i
- ☒ math.Sqrt
- ☐ if number < 2
- ☐ i++

ketikan dipanggil factorial(4) maka return value yang diberikan adalah \*

```
func factorial(n int) int {  
    if n == 1 {  
        return 1  
    } else {  
        return n * factorial(n - 1)  
    }  
}
```

- ☒ 24
- ☐ 10
- ☐ 120
- ☐ 15

Yang benar mengenai fungsi recursive adalah, kecuali \*

- ☒ memerlukan break poin untuk perulangan
- ☐ banyak masalah dapat diselesaikan dengan mudah dan code yang sedikit
- ☐ Fungsi yang memanggil dirinya sendiri
- ☐ harus memiliki return value

Algoritma berikut merupakan algoritma sorting jenis apa \*

```
count := make([]int, k + 1)
for i := 0; i < len(elements); i++ {
    count[elements[i]]++
}
counter := 0
for i := 0; i < k + 1; i++ {
    for j := 0; j < count[i]; j++ {
        elements[counter] = i
        counter += 1
    }
}

return elements
```

- ☒ Counting sort -  $O(n + k)$
- ☐ Selection sort -  $O(n^2)$
- ☐ Buble Sort
- ☐ Merge sort -  $O(\log n)$

Algoritma berikut merupakan algoritma pencarian jenis apa \*

```
for i := 0; i < len(elements); i++ {
    if elements[i] == x {
        return i
    }
}
return -1
```

- ☒ Linier search -  $O(n)$
- ☐ Binary Search
- ☐ Builtins Search
- ☐ Buble Search

Submit

Clear form

GoogleForms

This form was created inside of alterra.

