نمونه حل شده برای مثبتان

نشان دهید پیچیدگی الگوریتمی که زمان اجرای ان از رابطه ی T(n) = 2T(n - 1) به دست می آید، از مرتبه ی O(2^n) است.

▼ نمایش پاسخ

می دانیم که:

```
1) T(n) = 2T(n - 1)
```

است. حال به جای n در رابطه یک، n - 1 قرار می دهیم؛ در این صورت خواهیم داشت:

```
2) T(n - 1) = 2T((n - 1) - 1) = 2T(n - 2)
```

```
2^1 = 2
```

از جایگذاری دو رابطه 1 و 2 به رابطه رو به رو می رسیم:

```
3) T(n) = 2T(n - 1) = 2(2T(n - 2)) = 4T(n - 2)
```

```
2^2 = 4
```

حال به جای n در رابطه یک، n - 2 بزاریم و به رابطه رو به رو می رسیم:

```
4) T(n - 2) = 2T((n - 2) - 1) = 2T(n - 3)
```

که با جایگذاری دو رابطه 3 و 4 به رابطه زیر دست پیدا می کنیم:

```
T(n) = 4T(n - 2) = 4(2T(n - 3)) = 8T(n - 3)
```

```
2^3 = 8
```

اگر محاسبات بالا را همین طوری ادامه دهیم تا به جایگذاری n - n برسیم به عبارت

```
T(n) = 2^nT(n - n) = 2^n * T(0)
```

می رسیم که می دانیم T(0) از مرتبه 1 است.

پس داریم:

```
O(T(n)) = 2^n * 1 = 2^n
```

▼ نمونه الگوریتم

```
function towerOfHanoi(n, from_rod, to_rod, aux_rod)
{
    if (n == 0) {
        return;
    }
    towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
    print(n, from_rod, to_rod);
    towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
}
```

https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/