

ניתוח סיבוכיות

```
void Stage1(int n) {  
    for(int i=0; i<n; i++) { System.out.println("Operation"); }  
}
```

```
void Stage2(int complexity) {  
    for(int i=0; i<complexity; i++) { System.out.println("Operation"); }  
}
```

```
void Stage3(int n) {  
    for(int i=n; i>0; i--) { System.out.println("Operation"); }  
}
```

```
void Stage4(int n) {  
    for(int i=0; i<n; i++) {  
        for(int j=0; j<n; j++) {  
            // Basic Operation 1  
            // Basic Operation 1  
        }  
    }  
}
```

```
void Stage5(int n) {  
    for(int i=n; i>0; i--) {  
        for(int j=0; j<n; j++) {  
            // Basic Operation 1  
        }  
    }  
}
```

```
void Stage4dot1(int n) {  
    for(int i=0; i<n; i++) {  
        for(int j=0; j<i; j++) {  
            // Basic Operation 1  
        }  
    }  
}
```

```
int f(int i) { return i + 1; }  
void Stage6(int n) {  
    for(int i=0; i<n; i=f(i)) { System.out.println("Operation"); }  
}
```

```
int g(int i) { return i * 2; }  
void Stage7(int n) {  
    for(int i=0; i<n; i=g(i)) { System.out.println("Operation"); }  
}
```

```
void Stage8(int n) {  
    for(int i=1; i<n; i=g(i)) { System.out.println("Operation"); }  
}
```

```
int g2(int i) { return i * 2020; }  
void Stage8dot1(int n) {  
    for(int i=1; i<n; i=g2(i)) {  
        for(int zvi = 0; zvi < 2021; zvi++) System.out.println("Operation");  
    }  
}
```

```
int k(int i) { return i * (int)(  
    // Note: [0,1)*8 + 1 -> [0,8) + 1 -> [1,9)  
    Math.random()*8 + 2); }  
void Stage9(int n) {  
    for(int i=1; i<n; i=k(i)) { System.out.println("Operation"); }  
}
```

```
int p(int i) { return (int) Math.pow(i,2); }  
void Stage10(int n) {  
    for(int i=5; i<n; i=p(i)) { System.out.println("Operation"); }  
}
```

```
void Stage11(int n) {  
    // for(int i=0; i<n; i++) {  
    for(int i=1; i<=n; i*=2) {  
        for(int j=0; j<n; j++) {  
            // Basic Operation  
        }  
    }  
}
```

```
void Stage12(int n) {  
    // for(int i=0; i<n; i++) {  
    for(int i=1; i<=n; i*=2) {  
        for(int j=0; j<i; j++) {  
            // Basic Operation  
        }  
    }  
}
```

נשים לב כי i מקבל ערכים $1, 2, 4, \dots, n$, לפיכך, נגדיר משתנה k שרוץ מ-0 עד $\log n$, נגדיר את i להיות 2^k ונקבל:

$$T(n) = \sum_{k=0}^{\log n} \sum_{j=1}^i 1 = \sum_{k=0}^{\log n} \sum_{j=1}^{2^k} 1 = \sum_{k=0}^{\log n} 2^k = 2^{\log n + 1} - 1 = 2 \cdot 2^{\log n} - 1 = 2n - 1 = \theta(n)$$

עד עכשיו:

```
// [V]: for(int x=1; x < n; x*=3)
// [V]: for(int x=n; x > 1; i/=3)
// [V]: for(int x=2; x < n; x = Math.pow(x,3))
// [V]: for(int x=n; x > 2; x = Math.pow(x,1/3))
```

```
void Stage13(int n) {
    int x = 1;
    while (x < n) {
        x = x * 3;
    }
}
```

```
void Stage14 (int n){
    int x = n; // Only change this
    while (x > 1) {
        x = x / 3; // Same
    }
}
```

```
void Stage15(int n){
    int x = 2; // Inf if n > 1
    while (x < n) {
        x = x * x * x;
    }
}
```

```
void Stage16(int n){
    int x = 2; // This changed
    while (x < n) {
        x = x * x * x;
    }
}
```

```
void Stage17(int n){
    double x = n; // Only change this
    while (x > 2) {
```

```
x = Math.pow(x,1/3);  
}  
}
```

המרות:

```
forInit;  
while(Expression) {  
    Statement  
    forUpdate;  
}
```

```
for(forInit; Expression; forUpdate) { Statement }
```

ולכן:

```
// Stage13: for(int x=1; x < n; x*=3)  
// Stage14: for(int x=n; x > 2; i/=3)  
// Stage15: for(int x=1; x < n; x = Math.pow(x,3))  
// Stage16: for(int x=2; x < n; x = Math.pow(x,3))  
// Stage17: for(int x=n; x > 2; x = Math.pow(x,1/3))
```

```
void Stage18(int n) {  
    for(int i=2; Math.pow(2,i) < n; i++) {}  
}
```