

```
void proz3(int n)
{
    for (int a=1; a<=n; a++)
        for (int b=a; b<=n; b++)
            tuwas();
}
```

```
void proz2(int n)
{
    tuwas();

    if (n > 0){
        proz2(n-1);
    }

    tuwas();
}
```

- Berechnen Sie die Anzahl der Aufrufe von `tuwas()` für $n=4$.
- Bestimmen Sie die Anzahl der Aufrufe von `tuwas()` als Funktion von n .
- Bestimmen Sie die asymptotische Zeitkomplexität in O -Notation unter der Annahme, dass die asymptotische Zeitkomplexität von `tuwas()` $O(1)$ ist.

- b) Programmieren Sie die Funktion `fRekursiv` in Java, so dass Sie eine Zeitkomplexität von $O(2^n)$ und eine Rekursionstiefe von $O(n)$ besitzt.

```
public static int fRekursiv1(int n)
{
    assert (n>=1);
```

- Listen:

- 39 : A3 / c
- 38 : A3 / priorität
- 36 : A3 / b
- 35 : A3 / komplett
- 34 : A3 / generics
- 32 : A3 / bdc

- Bäume:

- 39 : A4 / c (iterativ?)
- 38 : A4 / d
- 37 : A4 / höhe rekursiv
ist binär iterativ?
- 36 : A4 / a+b / iterativ &
rekursiv
- 35 : A4 / komplett