

Aufgabe 2

Operationen: $+$, $-$, $<$, $>$, $\text{int } x$

1. Der Algorithmus muss ein exponentielles Wachstum aufweisen:

```
1 int main() {
2   int n = 5;
3   for (int i=0; i<=n; i++) {
4     i+=i;
5     for (int j=0; i<=n; i++) {
6       j+=j;
7     }
8 }
```

Hier benötigt jeder for-loop allein

$$3 \cdot \log n + 1$$

($i \leq n$, $i++$, $i+=i$) ($\text{int } i=0$)

Operationen, und ist damit $O(\log n)$

Werden nun zwei dieser Loops verschachtelt, so erhalten wir
 $O(\log n) \cdot O(\log n) = O((\log n)^2)$

2.

```
int main(){
  int n = 10;

  for (int i=0; i<n; i++){           // linear to n
    for (int j=0; j<n; j++){         // linear to n
      for (int k=0; k<n; k++){       // log(n)
        k+=k;
      }
    }
  }
  // in total n * n * log(n) = n^2 * log(n)
}
```

3.

```
1 int subtraction(int n) {
2   if (n == 0) return 0;
3   else for (n; n>=1; n--) subtraction(n-1);
4   return 0;
5 }
```

Ruft die Rekursion n mal auf.
Das führt zu $2^n - 1$ Rechenschritten