Ex 9,7)

```
def h(V):
    n = len(V)                          -> O(1)
    for i in range(n):                  -> O(n) O(log n)
        V[i] = 0
        j = n
        while j > 1:                    -> O(log_2 n) = log n
            V[i] += i * j
            j //= 2
```

Ex 9,8)

```
def fuse(t1, t2)                        -> O(n1^2 + n2^2)
    if len(t1) == 0:
        return t2
    elif len(t2) == 0:
        return t1
    elif t1[0] < t2[0]:
        return [t1[0]] + fuse(t1[1:], t2)
    else:
        return [t2[0]] + fuse(t1, t2[1:])
```

$O(n_1^2)$ = $n_1 + (n_1 - 1) + (n_1 - 2) \cdots$

$n_2 + (n_2 - 1) + (n_2 - 2) \cdots$