AD 5

```
public int dequeue() {
    if (S2.isEmpty())                    O(1)
        while (!st.isEmpty())            O(n)
            s2.push(st.pop());           O(1)
        return S2.pop();                 O(1)
}                                        O(n)
```

GADS 1

(5) ~~(a) if (s2.is~~

(b) Ich definiere, dass für jedes enqueu und dequeue man +4 Punkte bekommt und für jede Operation verbraucht man einen Punkt. Also nach einem enqueue bekommt man immer +3 eingezahl auf das Konto. Und bei dequeue gibts zwei Fälle:

1. s2.isEmpty() = false

   Dann bekommt man +2 am Ende.

2. !s2.isEmpty()

   Dann wird die Schleife n-mal ausgeführt:

$-3n - 2 + 4 = \underline{-3n + 2}$.

Betrachten wir den worst case: es wurde n-mal enqueued und dann will man dequeue machen:

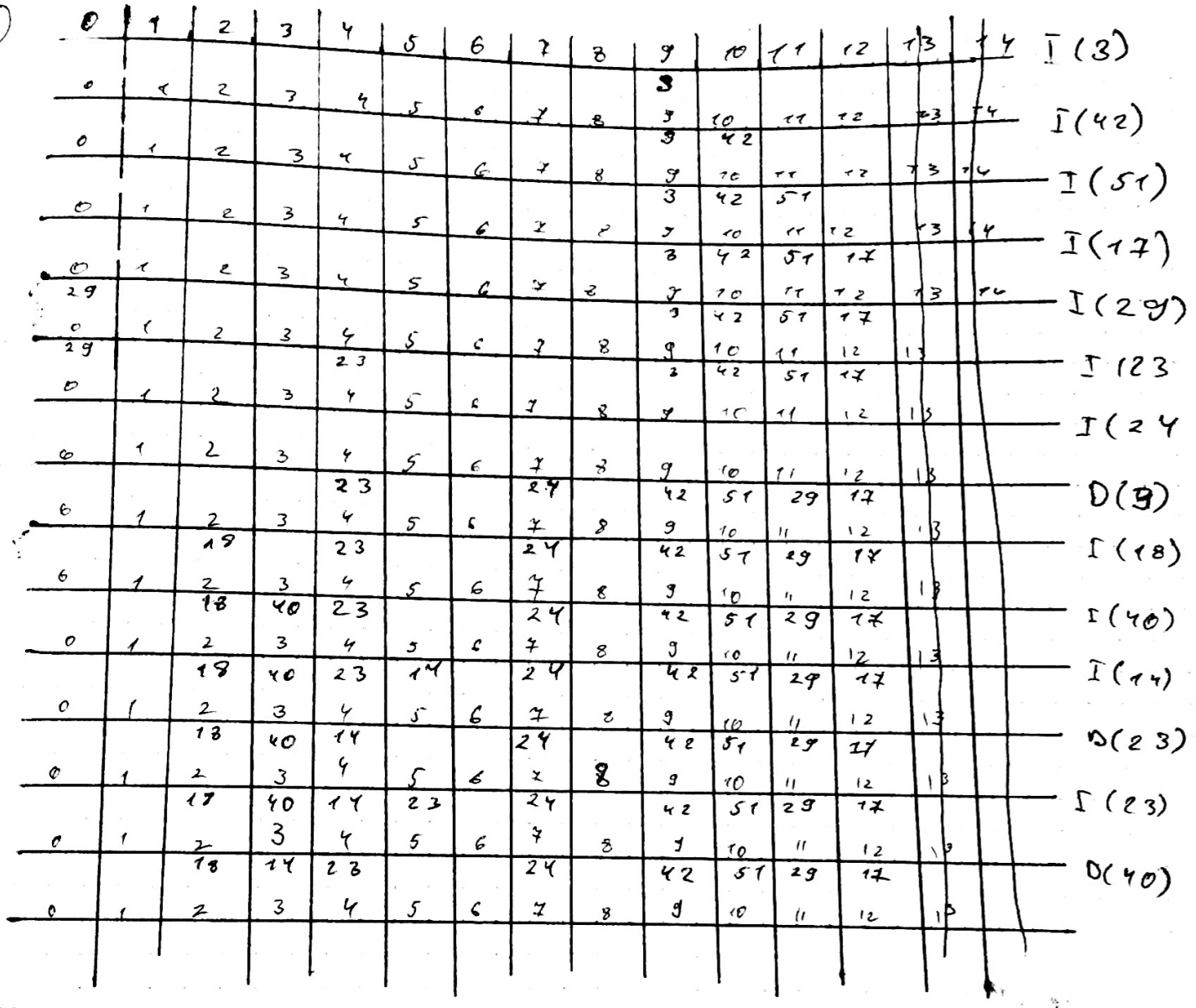
$3n - 3n + 2 = 2 > 0$

Also es das Bankkonto ist immer Positiv.

$A(\text{enqueue}) = O(1) + O(3) = O(1)$

$A(\text{dequeue}) = 3n + 2 - 3n + 2 = 4 = O(1)$

Nun hat man immer ein const daufrent


I (3)
I (42)
I (51)
I (17)
I (29)
I 123
I (24
D(9)
I (18)
I (40)
I (14)
D(23)
I (23)
D(40)