

---

---

# ÜBUNGSSTUNDE 16.4.14

Datenstrukturen und Algorithmen - Daniel Graf





# Quersumme

- Aufgabe: Zähle die Anzahl  $n$ -stelliger Zahlen mit (nicht-rekursiver) Quersumme exakt  $m$
- Beispiel:  $n=2$ ,  $m=17$ :
  - 2 Möglichkeiten: 89, 98
- Tipps auf dem Aufgabenblatt:
  - Dynamische Programmierung
  - erwartete Laufzeit:  $O(n^2)$   
(soll damit auch für Speicher gelten)



# Quersumme

- $T[n][m]$  = Anzahl Möglichkeiten Zahlen mit  $n$  Ziffern und Quersumme  $m$  zu bilden
- Basis:  $T[1][i] = 1, \forall 1 \leq i \leq 9$
- Schritt:  $T[n][m] = \sum_{i=0}^{\min\{9,m\}} T[n-1][m-i]$
- Bemerkung:  $T[n][m] = 0$  falls  $m > 9n$
- DP-Tabelle nur einmal berechnen und für alle Anfragen benutzen



# Quersumme

```
import java.util.Scanner;

class Main {
    static Scanner scanner;
    static int [][] T;
    static int MODN = 1000000000+9;

    private static void build_table() {
        int N = 1000;
        int M = 9*N;
        T = new int[N+1][M+1];
        for(int i=1; i<=9; i++)
            T[1][i] = 1;
        for(int n=2; n<=N; n++) {
            for(int s=0; s<=M; s++) {
                for(int i=0; i<=Math.min(9,s); i++) {
                    T[n][s] = (T[n][s]+T[n-1][s-i])%MODN;
                }
            }
        }
        T[1][0]=1;
    }
    ...
}
```



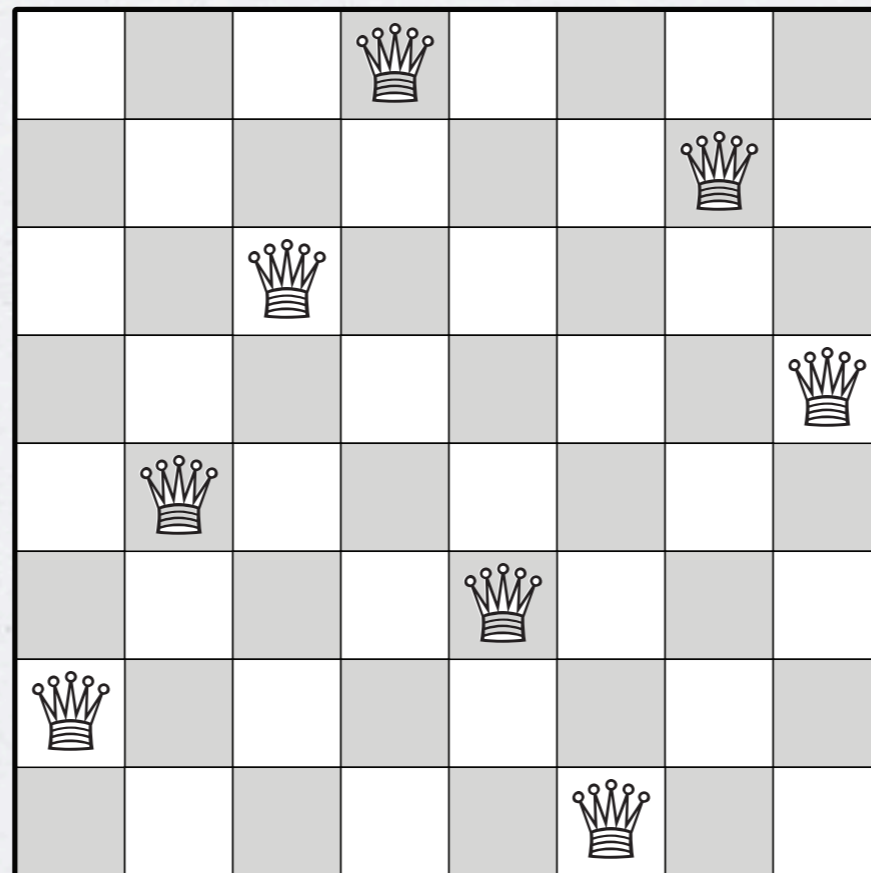
# Quersumme

```
...
private static void testcase() {
    int N = scanner.nextInt();
    int M = scanner.nextInt();
    if(M>9*N)
        System.out.println(0);
    else
        System.out.println(T[N][M]);
}

public static void main(String[] args) {
    build_table();
    scanner = new Scanner(System.in);
    long T = scanner.nextLong();
    for(long t=0; t<T; t++) testcase();
    scanner.close();
}
}
```

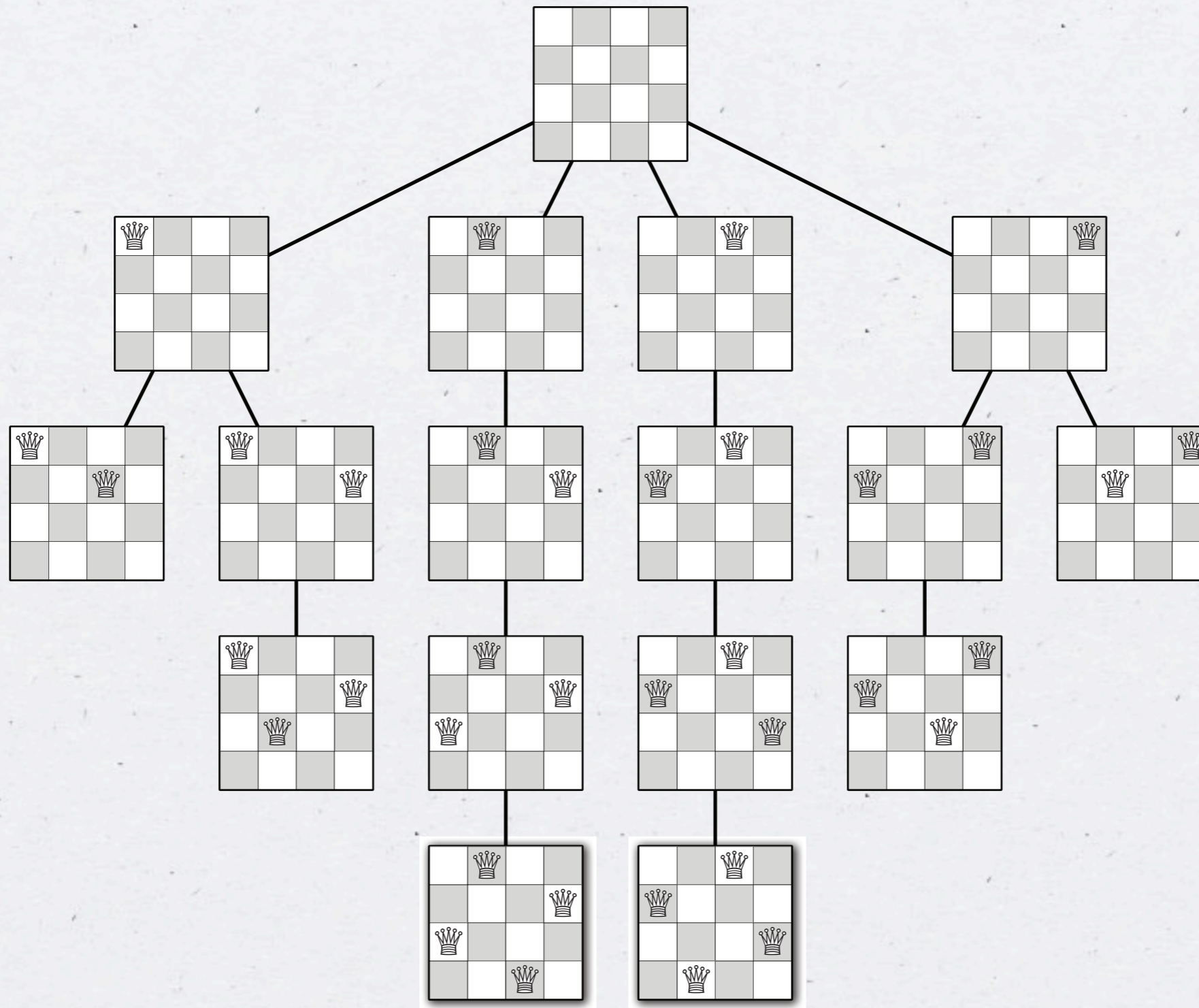
# DP vs. Backtracking

- Hürde beim Damenproblem: exponentielle Anzahl Teilprobleme, die sich nur wenig überlappen
- Backtracking: systematisches Durchprobieren

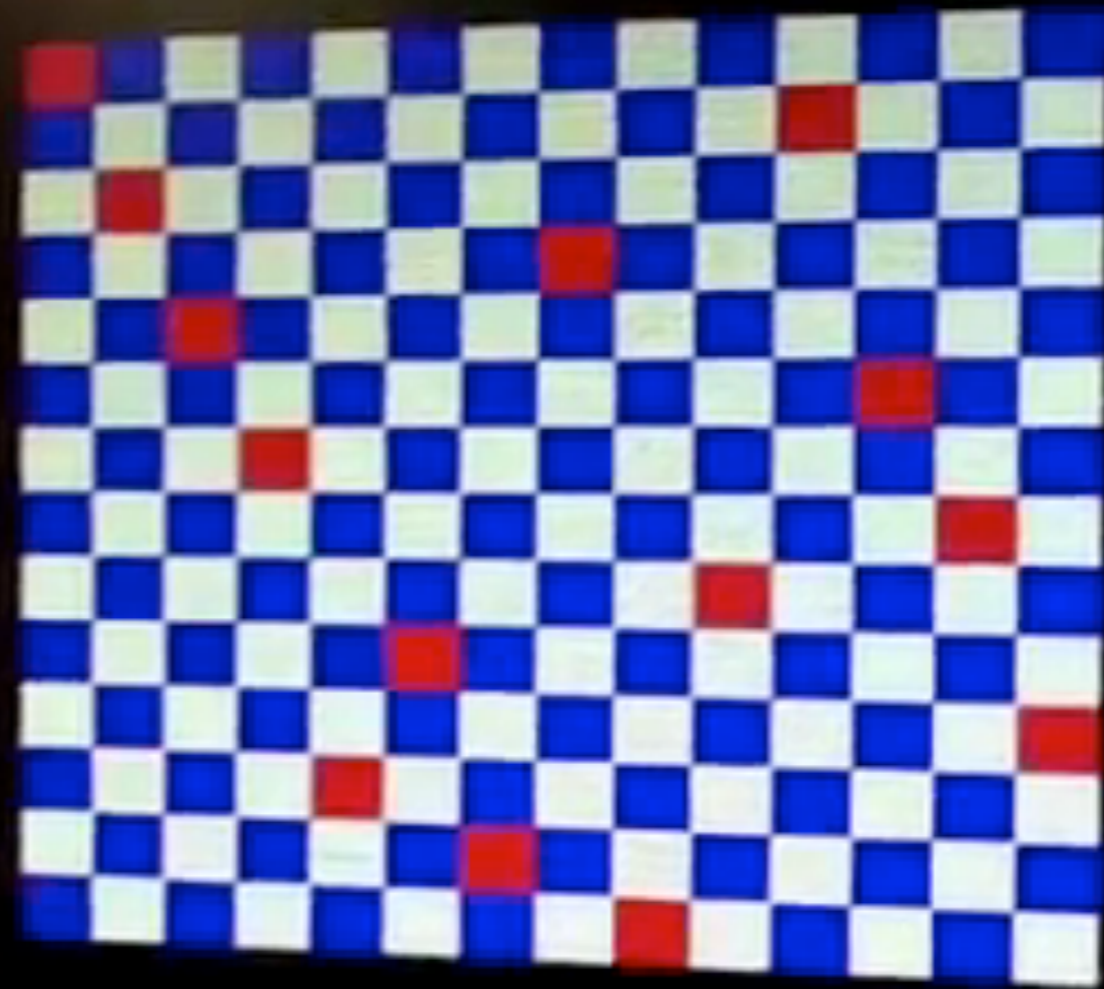




# DP vs. Backtracking



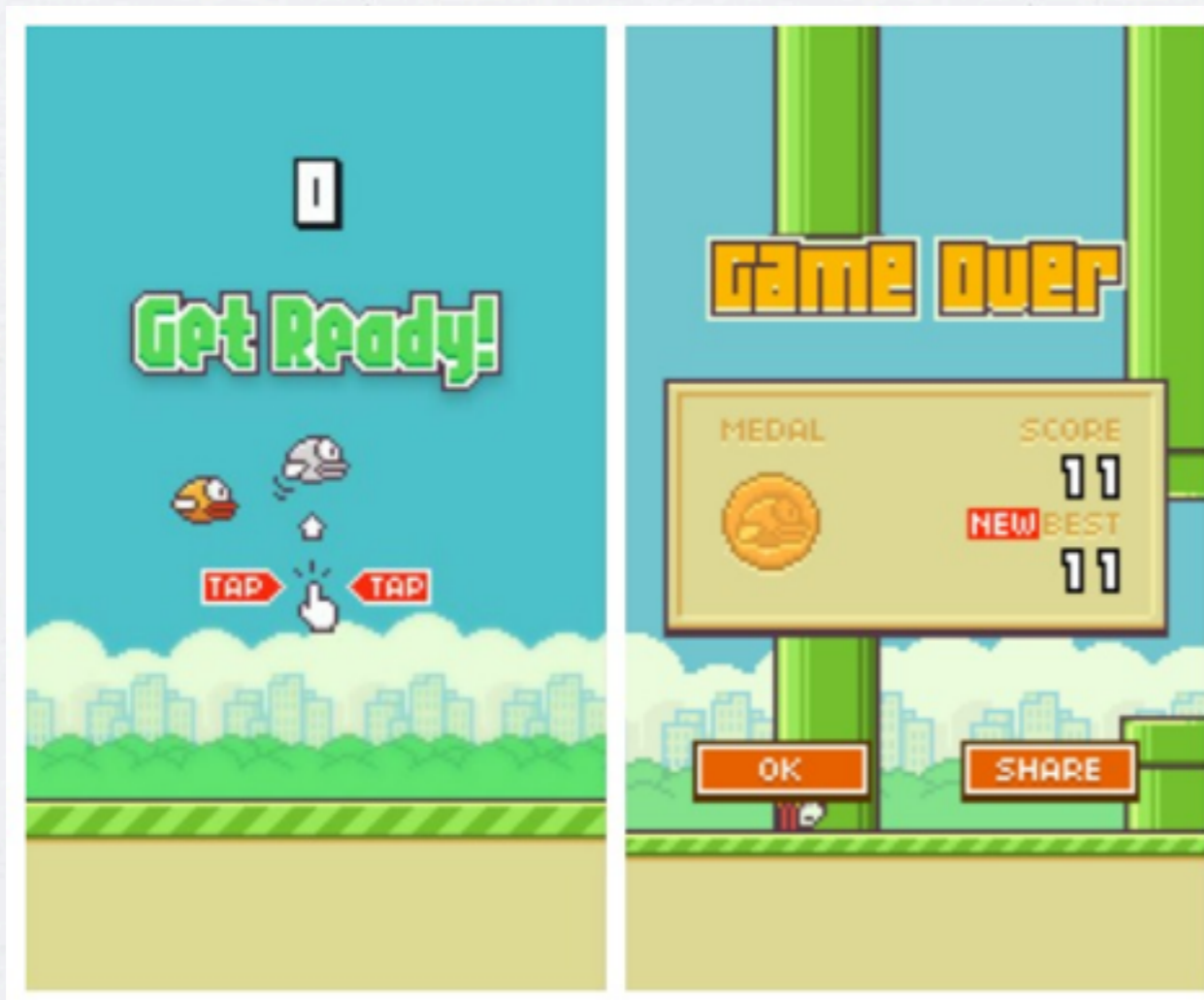
# D&A meets Digitech





# Flappy Bird

- Aufgabe: Finde Flügelschlag-Sequenz um von links nach rechts zu fliegen ohne ein Hindernis zu berühren



```
|||||  
00|00000|100|10|  
|0000|10|00|00||  
|000000|100|111|  
|0|0000|00|100|0  
000|0|10|1000|00  
0000000|0|00|11|0  
|00000|000000|00  
0000000|00|00|00  
|||||
```

> <http://dgraf.ch/da14>



*Frohe Ostern*