

Algorithmen & Datenstrukturen

Woche 7

Marius Tomek, Nicolas Wehrli, Tim Rieder

7. November 2022

ETH Zürich

Kurze Kommentare zur letzten Serie

Dynamische Programmierung

Homework 06

New groups & Peergrading 6.4

Kurze Kommentare zur letzten Serie

Häufige Fehler:

- 5.3b: $\mathcal{O}(\log \log n)$
- 5.5: k_{11} oder k_{12} , 3 oder 4 Child-nodes

5min Zeit, Korrekturen anschauen

Dynamische Programmierung

- Subset sum
- Knapsack
- Knapsack approximative (Idee & dann viel rumrechnen)
- Matrix chain multiplication

Homework 06

Exercise 6.4

```
1 Input: int m, n. Grid A
2
3 function f(x,y)
4     if x>m or y>n or A[x][y]=="#" then return 0
5     coinHere = 1 if A[x][y]=="C", 0 otherwise
6
7     return coinHere + max{f(x+1,y), f(x,y+1)}
```

Pseudocode 6.4a)

Exercise 6.4

```
1 Input: int m, n. Grid A // 6.4 (d)
2
3 function f(x,y)
4   dp <- int array of [1...m+1]x[1...n+1]
5   for i=1...m+1 begin dp[i][n+1] = 0 end
6   for j=1...n+1 begin dp[m+1][j] = 0 end
7
8   for i=m...1 begin
9     for j=n...1 begin
10      if A[i][j]=="#" then dp[i][j] = 0; do next iteration
11
12      coinHere = 1 if A[i][j]=="C", 0 otherwise
13      dp[i][j] = coinHere + max{dp[i+1][j], dp[i][j+1]}
14    end
15  end
16  return dp[1][1]
```

Pseudocode 6.4 d)

New groups & Peergrading 6.4
