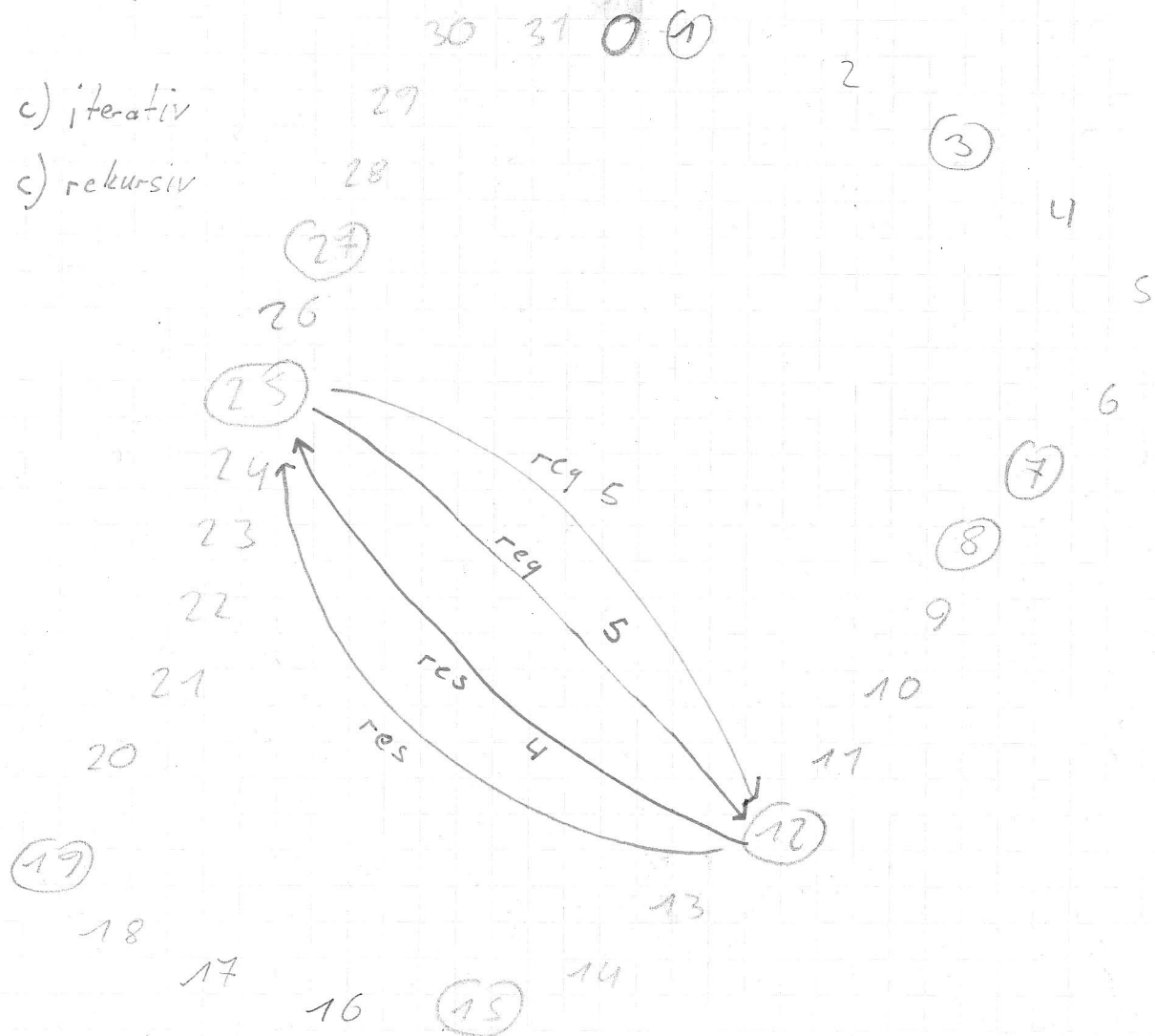


1. a)

c) iterativ

c) rekursiv



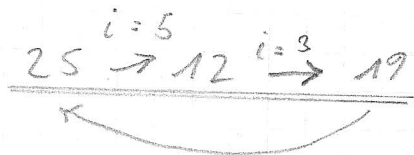
FT 1: $FT_p[i] = \text{succ}(p + 2^{i-1})$

FT 2S:

i	start	Int.	succ
1	2	[2, 3)	3
2	3	[3, 5)	3
3	5	[5, 9)	7
4	9	[9, 17)	12
5	17	[17, 1)	19

i	start	Int.	succ
1	26	[26, 27)	27
2	27	[27, 29)	27
3	29	[29, 1)	1
4	1	[1, 9)	1
5	9	[9, 25)	12

b) Message from 25 to 19:



FT 12:

	start	Int.	succ
1	13	[13, 14)	15
2	14	[14, 16)	15
3	16	[16, 20)	19
4	20	[20, 28)	25
5	28	[28, 1)	1

d) yes; callback as usual!

e) Iterativ: Source kann den Weg mitverfolgen

Rekursiv: weniger Aufwand für Source Node

f) $O(\log n)$

worst case: $\log_2 n$ $(d(32) = 5)$

- 2) a) 1. find position $\rightarrow 21$, find successor $\rightarrow 25$
 2. fill finger table

FT 21:

	start	int.	succ
1	22	[22, 23)	25
2	23	[23, 25)	25
3	25	[25, 29)	25
4	29	[29, 5)	1
5	5	[5, 21)	7

3. update other finger tables

1. $i = 1$

$p = 19 = \text{lookup}_p(20)$

$25 = p.\text{get_finger}(1) > 21$

17. $\text{set_finger}(1, 21)$

$p = 15 = \text{get}_p()$

$15.\text{get_finger}(1) = 19 > 21 ? = \text{false}$

$i = 2$

$i = 5$

b) FT: 8 succ = 12, pre = 7

	start	int	succ
1	9	[9, 10)	12
2	10	[10, 12)	12
3	12	[12, 16)	12
4	16	[16, 24)	19
5	24	[24, 8)	24

19 $\xrightarrow{\text{pre}}$ 15 $\xrightarrow{\text{pre}}$ 12 (failed) $\Rightarrow \text{succ}(8) = 15$

\Downarrow

finger table update von 12 rückwärts!

pre(15) = 8

c) überall selbst eingetragen (finger table, succ, pre)

3) a) find node resp. 16

1 \rightarrow find largest id $\leq 16 = \text{find pre}(16) = 15 \rightarrow \text{succ}(15)$
 $= 19 \rightarrow \text{save object}$

1 \rightarrow find largest id $\leq 21 = \text{find pre}(21) = 19 \rightarrow \text{succ}(19) =$
 $= 25 \rightarrow \text{save object}$

b) join 21 : $[19, 21]$

- c) i. doppelt speichern in FT Eintrag 1 (succ)
ii. doppelt speichern in FT Eintrag m (gegenüber)