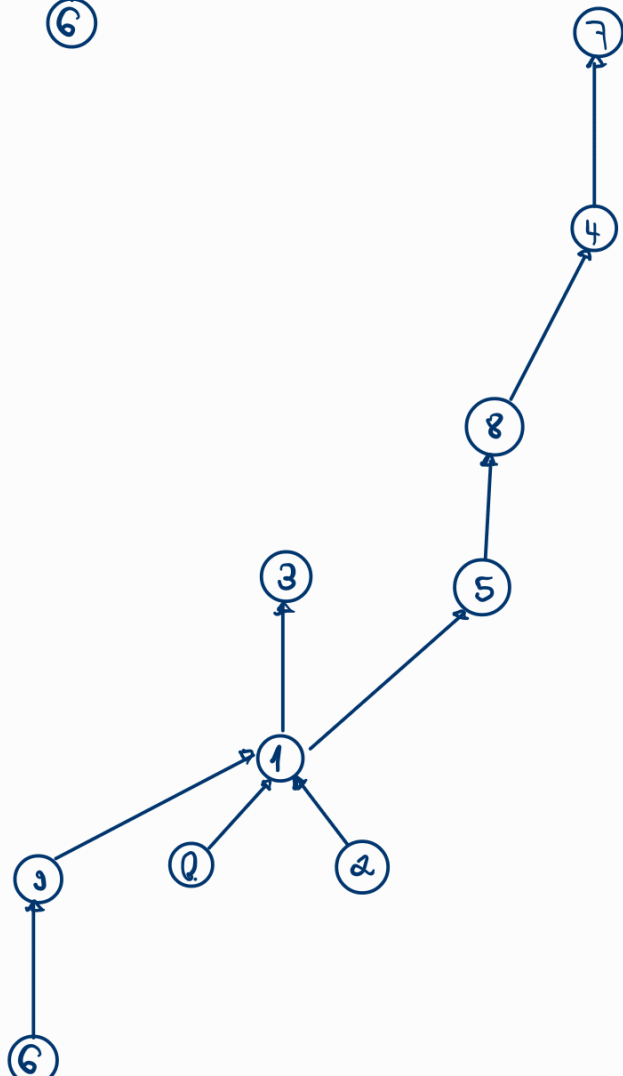
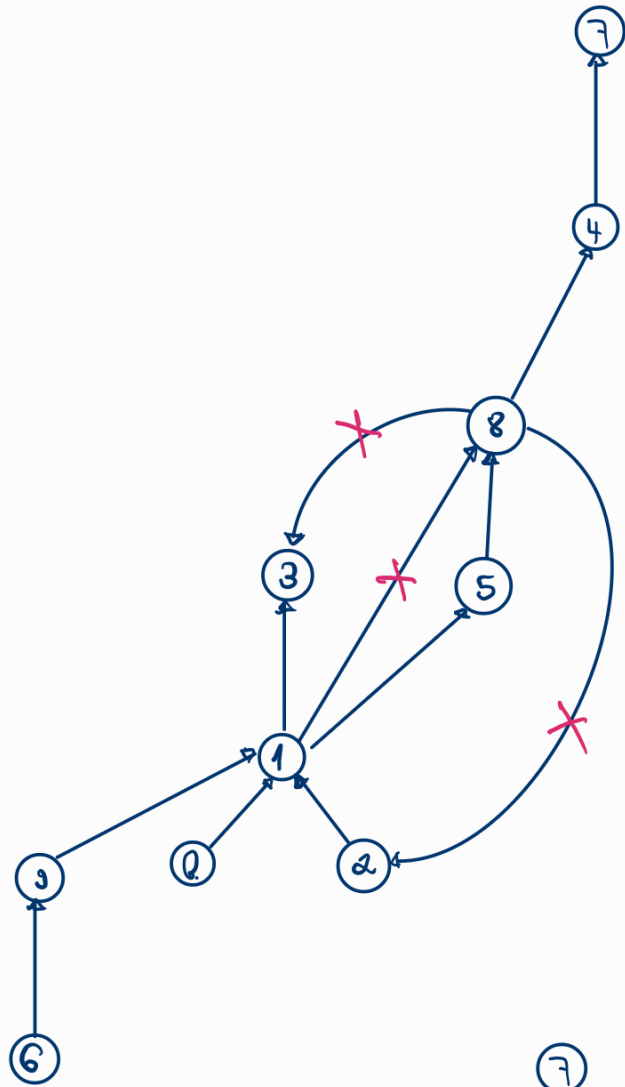


2 B 1.)

0	1
2	1
1	3
6	9
4	7
5	6
1	5
1	8
9	1
8	2
8	3
8	4



2.) Lari: "Lineares Wachstumsverhalten"

Fabi: "Kommt d'rauf an. Es schwankt zwischen konstant und schlimmstenfalls quadratisch."

3.)

Quick-Union-Algorithmus

0	1
2	1
1	3
6	9
4	7
5	6
1	5
1	8
9	1
8	2
8	3
8	4

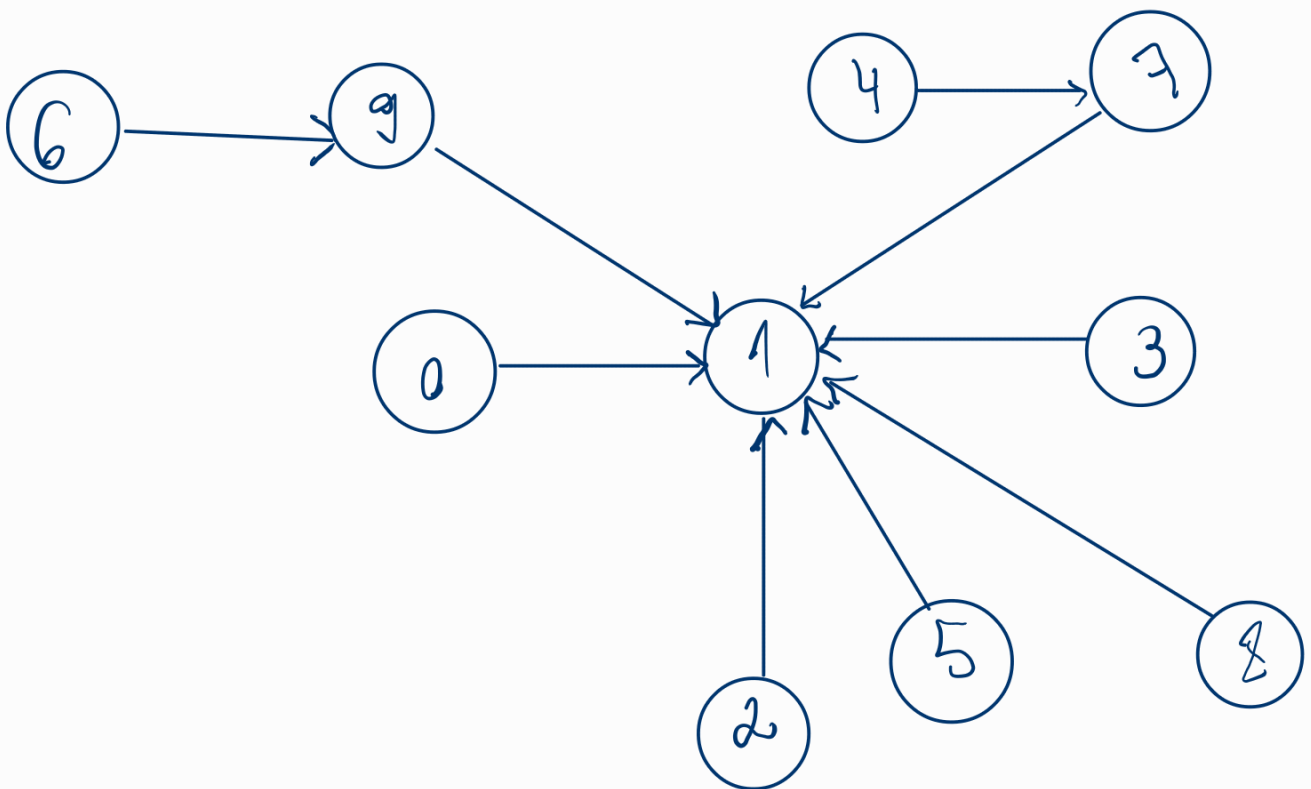
0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
1	1	1	3	4	8	6	7	8	9
1	3	1	3	4	5	6	7	8	9
1	3	1	3	4	5	9	7	8	9
1	3	1	3	4	5	9	7	8	9
1	3	1	3	4	8	9	7	8	9
1	5	1	3	4	8	9	7	8	9
1	8	1	3	4	8	9	7	8	9
1	8	1	3	4	8	9	7	8	1
1	8	1	3	4	8	9	7	2	1
1	8	1	3	4	8	9	7	3	1
1	8	1	3	4	8	9	7	4	1

Weighted Quick-Union

0	1
2	1

0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9

1 3	3 1	1 1 1 3 4 5 6 7 8 9
6 9		1 1 1 1 4 5 6 7 8 9
4 7		1 1 1 1 4 5 9 7 8 9
5 6	5 1	1 1 1 1 7 5 9 7 8 9
1 5	8 1	1 1 1 1 7 8 9 7 8 9
1 8	2 8	1 1 1 1 7 1 9 7 8 9
9 1	3 8	1 1 1 1 7 1 9 7 1 9
8 2	7 1	1 1 1 1 7 1 9 7 1 1
8 3		:
8 4		1 1 1 1 7 1 9 1 1 1



4.)

Logarithmisch

da die Kette bis zur Wurzel durchlaufen wird
Schlimmstenfalls alle Knoten in einer Kette

6.)

Logarithmisch

da die Kette bis zur Wurzel durchlaufen wird
Schlimmstenfalls alle Knoten in einer Kette

2 b

1. Weil jedes Element mit jedem anderen verglichen werden muss
2. Weil zwar nicht $all\ rs\ all$, aber gewähltes Element gegen alle schon sortierten
 \Rightarrow gegen Ende Schlimmstenfalls $all\ rs\ all$

3. VIRENBEFALL

selection:

[] [VIRENBEFALL]

[A] [VIRENBEFLL]

[AB] [VIRENEFLL]

[ABE] [VIRNEFLL]

[ABEE] [VIRNELL]

[ABEEF] [VIRNLL]

[ABEEFI] [VRNLL]

[ABEEFIL] [VRNL]

[ABEEFILL] [VRN]

INSERTION

[VIRENBEFALL]

[VIRENBEFALL]

[IRVENBEFALL]

[IERVNBEFALL]

[IENRVBEFALL]

[BIENRV EFALL]

[BI EENRV FALL]

[BIEEFNRVALL]

[ABIEEFNRV LL]

↑ 1 BESSER