Q1: Dijkstra $v_1 - v_4$ I

I. 1. $\ell(v_1) = 0$

$\ell(v_i) = \infty \quad \forall i \neq 1$

$R := \emptyset$

2. v_1

3. $R = \{v_1\}$

4. • $\underset{\infty}{\underset{\parallel}{\ell(v_2)}} > \ell(v_1) + c((v_1, v_2)) = 12 \Rightarrow \ell(v_2) := 12, \; p(v_2) := v_1$

• $\ell(v_6) = 4, \; p(v_6) = v_1$

• $\ell(v_7) = 2, \; p(v_7) = v_1$

• $\ell(v_8) = 6, \; p(v_8) = v_1$

5. $R \neq V(G)$

2. v_7

3. $R = \{v_1, v_7\}$

4. $4 = \ell(v_6) \not> \ell(v_7) + 9 = 11$

5. $R \neq V(G)$

2. v_6

3. $R = \{v_1, v_6, v_7\}$

4. • $\ell(v_5) = 7, \; p(v_5) = v_6$

5. $R \neq V(G)$

2. v_8

3. $R = \{v_1, v_6, v_7, v_8\}$

4. • $12 = \ell(v_2) \not> \ell(v_8) + 6 = 12$

• $\ell(v_9) = 15, \; p(v_9) = v_8$

5. $R \neq V(G)$

II

(2.) v_5

(3.) $R = \{v_1, v_5, v_6, v_7, v_8\}$

(4.) • $\ell(v_4) = 10$, $p(v_4) = v_5$

(5.) $R \neq V(G)$

(2.) v_4

(3.) $R = \{v_1, v_4, v_5, v_6, v_7, v_8\}$

(4.) /

(5.) $R \neq V(G)$

(2.) v_2

(3.) $R = \{v_1, v_2, v_4, v_5, v_6, v_7, v_8\}$

(4.) • $\ell(v_3) = 17$, $p(v_3) = v_2$

(5.) $R \neq V(G)$

(2.) v_9

(3.) $R = \{v_1, v_2, v_4, v_5, v_6, v_7, v_8, v_9\}$

(4.) $17 = \ell(v_3) \not> \ell(v_9) + 4 = 15$

(5.) $R = V(G)$

(2.) v_3

(3.) $R = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$

(4.) keine Knoten $w \in V(G) \setminus R$

STOP

length shortest path $v_1 - v_4$: 10

$v_1 - v_6 - v_5 - v_4$

Q2:

LOOPS:

(a) $k=0$

FOR $n=1$ TO 20

$k=k+1$

enters loop 20 times, each time k is incremented by 1

$\rightarrow k=20$ in the end

Q3:

(b) $k=0$

FOR $n=1$ TO 5

$k=(k+1)\cdot n$

$n=1$: $k=(0+1)\cdot 1=1$

$n=2$: $k=(1+1)\cdot 2=4$

$n=3$: $k=(4+1)\cdot 3=15$

$n=4$: $k=(15+1)\cdot 4=60$

$n=5$: $k=(60+1)\cdot 5=305$

$\rightarrow k=305$ in the end

IV

Q4:

Anna	Emil	Frank	Daniel
Bella	Frank	Daniel	Emil
Carla	Emil	Daniel	Frank

Daniel	Bella	Carla	Anna
Emil	Carla	Anna	Bella
Frank	Carla	Bella	Anna

(a)

GS: with women proposing:

- w = Anna is unmatched
  m = Emil is unmatched → (Anna, Emil) engaged
- w = Bella is unmatched
  m = Frank is unmatched → (Bella, Frank) engaged
- w = Carla is unmatched
  m = Emil
  m prefers w over his current partner w^c = Anna
  ↳ (Carla, Emil) engaged
  → Anna unmatched
- w = Anna is unmatched
  m = Frank
  m prefers his current partner w^c = Bella over w = Anna
  m = Daniel is unmatched → (Anna, Daniel) engaged

↳ We get pairs (Anna + Daniel, Carla + Emil, Bella + Frank

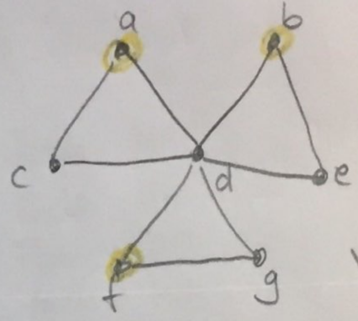
(b) Daniel suggests the following pairs Anna + Frank, Bella + Emil, Carla + Daniel $\square$
this is not stable, as Anna + Emil form a blocking pair, that is, they would both be happier by switching to this pair, than when sticking to they assigned partner

Scanned by CamScanner

V

Q5:

Greedy IS:



$V = \{a, b, c, d, e, f, g\}$

I. $I = \emptyset$

II. $V \neq \emptyset$

  Choose a, $I = \{a\}$

  $V = \{b, e, f, g\}$

 $V \neq \emptyset$

  Choose b, $I = \{a, b\}$

  $V = \{f, g\}$

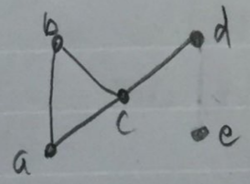
 $V \neq \emptyset$

  Choose f, $I = \{a, b, f\}$

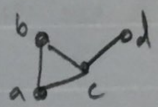
  $V = \emptyset$

III. Return $\{a, b, f\}$

RAMIS:



1. not empty
2. e is isolated vertex $0+1=1$, G:

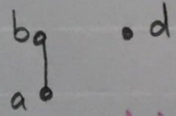


1. not empty, 2. no isolated vertex

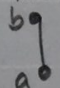
3. Choose c (max degree)

- c in IS $1+1=2$
  - ↳ Delete c and all neighbors
  - G: /
  - 1. G is empty
  - IS of size 2, $\{e, c\}$

- c not in IS
  - ↳ Delete c: G:



  1. not empty
  2. d is isolated vertex $1+1=2$
  G:



  1. not empty, 2. no isolated vertex

↓ (*)

(*) 3. Choose a (both a, b degree 2 → any will do)

- a in IS $2+1=3$
  - ↳ Delete a and its neighbors
  - G: /
  - 1. G is empty
  - IS of size 3 $\{e, d, a\}$

- a not in IS
  - ↳ Delete a
  - G: • b
  - 1. G not empty
  - 2. b is isolated vertex $2+1=3$
  - G: /
  - 1. G is empty
  - IS of size 3 $\{e, b, d\}$

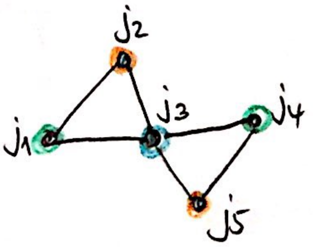
Q7: Chinese Postman Problem 4 too few outgoing

VI

Q6: Scheduling Conflicting Jobs:

j1	Frank	M1
j2	Frank	M1
j3	Frank	M2
j4	Jacob	M2
j5	Anna	M2

conflict graph:



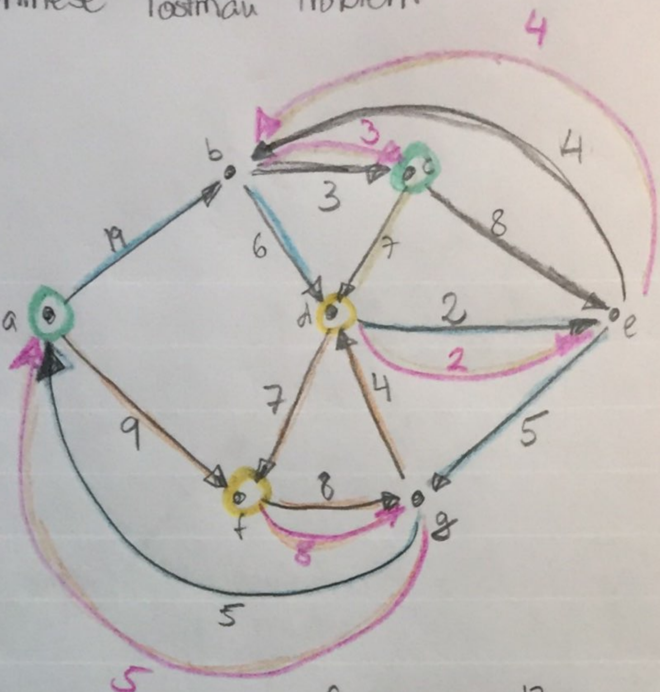
We need to find out how many days we need.
In a coloring, all vertices of a single color form an IS and can be executed simultaneously.
here ⇒ # colors $\hat{=}$ # days

⇒ we use Alg. 6.9 (guarantees us that we need at most 5 colors)

1. $c(j_i) = \infty \quad \forall i = 1, .., 5$
2. • j_1: $c(j_1) = \min(\mathbb{N}) = 1$
   • j_2: $c(j_2) = \min(\mathbb{N} \setminus \{1\}) = 2$
   • j_3: $c(j_3) = \min(\mathbb{N} \setminus \{1,2\}) = 3$
   • j_4: $c(j_4) = \min(\mathbb{N} \setminus \{3\}) = 1$
   • j_5: $c(j_5) = \min(\mathbb{N} \setminus \{1,3\}) = 2$

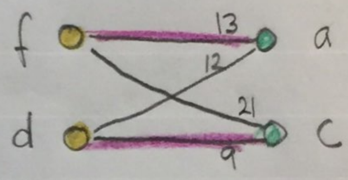
We used 3 colors → we can schedule the jobs in 3 days.

Q7: Chinese Postman Problem



too few outgoing edges
too few incoming edges

(a) for a, c, d, f indegree ≠ outdegree
→ no Eulerian cycle
→ can't use each street only once



min weight matching

(b) For all vertices now indegree = outdegree → Hierholzer algorithm

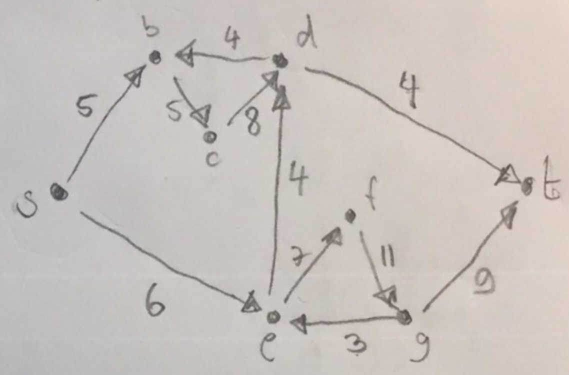
1. Choose a, C = a-b-d-e-g-a
2. not a Eulerian cycle
3. Delete all edges from C, choose b, C' = b-c-e-b
   ↳ C = a-b-c-e-b-d-e-g-a
2. not a Eulerian cycle
3. Delete all edges from C, choose a, C' = a-f-g-d-f-g-a
   ↳ C = a-f-g-d-f-g-a-b-c-e-b-d-e-g-a
2. not a Eulerian cycle
3. Delete al edges from C, choose c, C' = c-d-e-b-c
   ↳ C = a-f-g-d-f-g-a-b-c-d-e-b-c-e-b-d-e-g-a
2. C is a Eulerian cycle: STOP

length of tour = 22 (newly added edges) + (19+3+6+4+7+8+2+7+4+5+8+9+5) (original edges) = 109

Q8: (G, c, s, t):

flow from s to t

Flow:

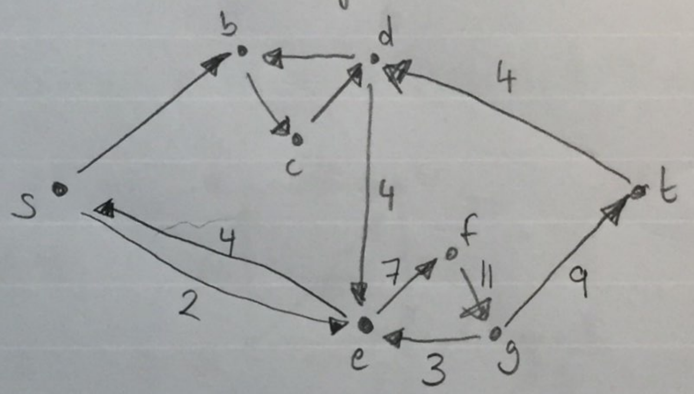


Edmonds Karp: flow of 0 on all edges

First residual graph, shortest s-t-path in G_f s-e-d-t, $\gamma = 4$

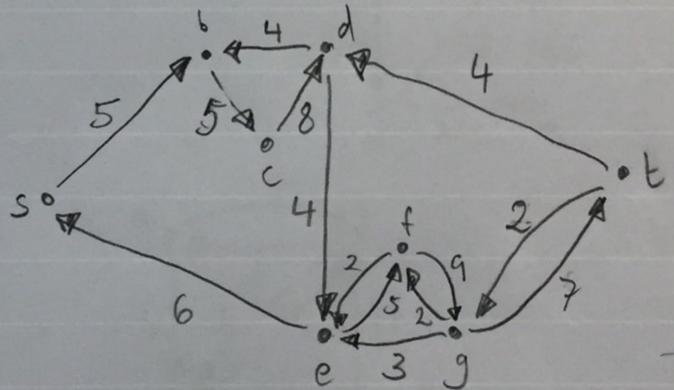
($\hat{=}$ min # edges) $\hat{=}$ f-augmenting path

new G_f:



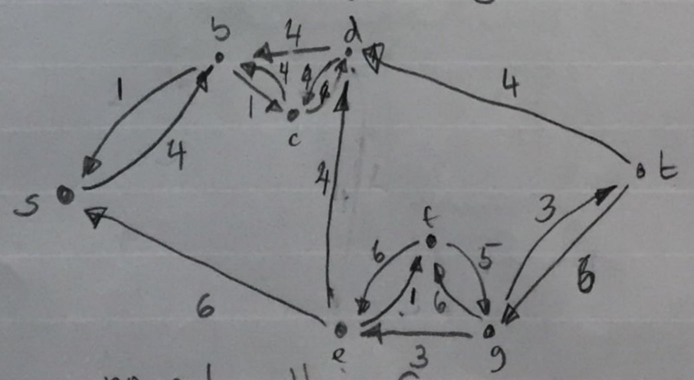
f-augmenting path s-e-f-g-t, $\gamma = 2$

new G_f:



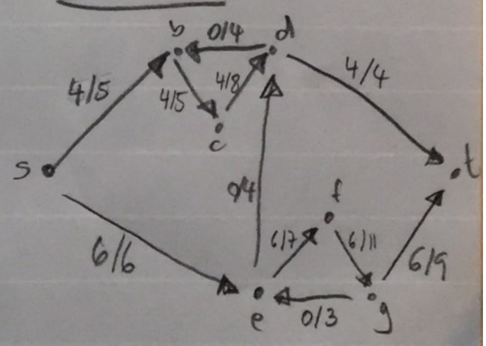
f-augmenting path s-b-c-d-e-f-g-t, $\gamma = 4$

new G_f:



no s-t-path in G_f → flow is optimal

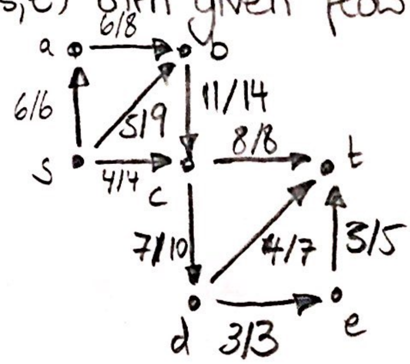
flow in G:



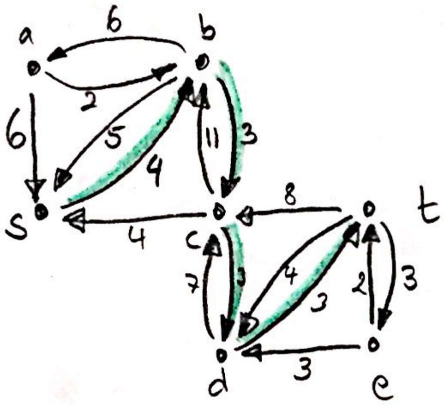
flow value: 10

Q9:

Network (G, c, s, t) with given flow:

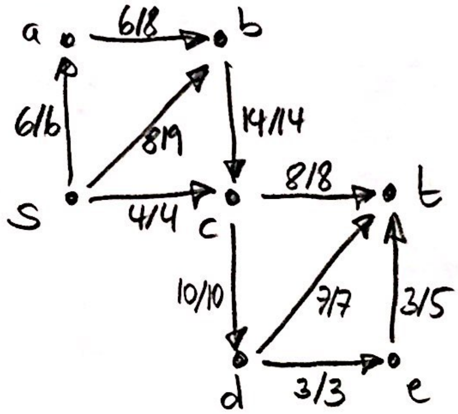


(a) G_f:



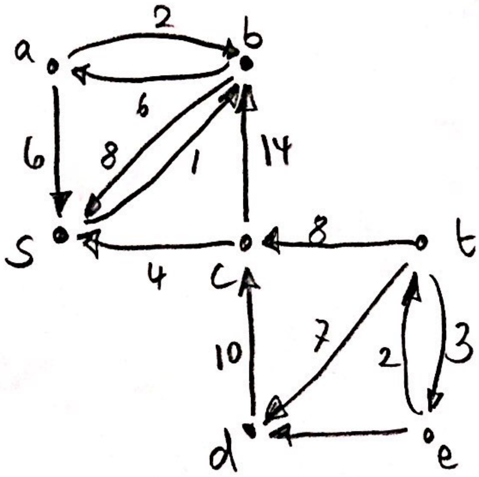
(b) f-augmenting path ≙ shortest (fewest edges) s-t-path in G_f:
$\gamma = 3$

new network:



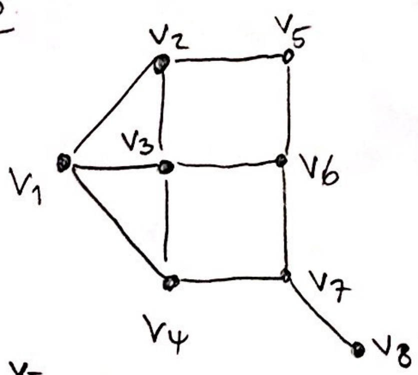
flow value: 18

(c) new G_f:

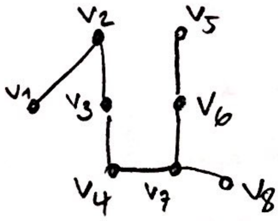


there exist no s-t-path in G_f
$\Rightarrow$ f is optimal

Q10: DFS/BFS:



DFS:



Q: V_1

$V_2 V_1$

$V_3 V_2 V_1$

$V_4 V_3 V_2 V_1$

$V_7 V_4 V_3 V_2 V_1$

$V_6 V_7 V_4 V_3 V_2 V_1$

$V_5 V_6 V_7 V_4 V_3 V_2 V_1$

$V_6 V_7 V_4 V_3 V_2 V_1$

$V_7 V_4 V_3 V_2 V_1$

$V_8 V_7 V_4 V_3 V_2 V_1$

$V_7 V_4 V_3 V_2 V_1$

$V_4 V_3 V_2 V_1$

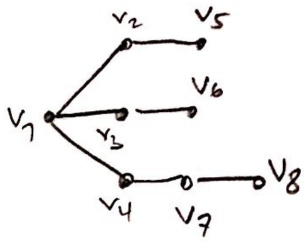
$V_3 V_2 V_1$

$V_2 V_1$

V_1

$\emptyset$

BFS:



Q: V_1

$V_1 V_2$

$V_1 V_2 V_3$

$V_1 V_2 V_3 V_4$

$V_2 V_3 V_4$

$V_2 V_3 V_4 V_5$

$V_3 V_4 V_5$

$V_3 V_4 V_5 V_6$

$V_4 V_5 V_6$

$V_4 V_5 V_6 V_7$

$V_5 V_6 V_7$

$V_6 V_7$

V_7

$V_7 V_8$

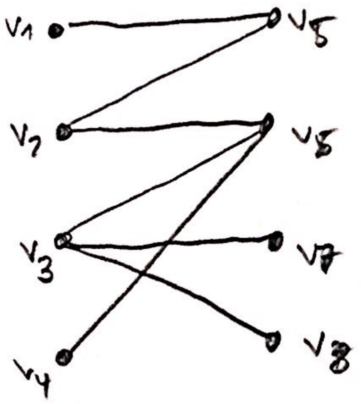
V_8

$\emptyset$

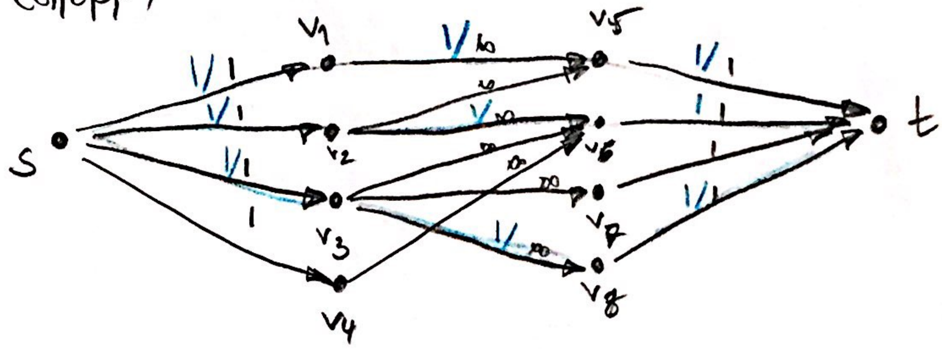
Q11:

Maximum Matching in bipartite graphs:

G:

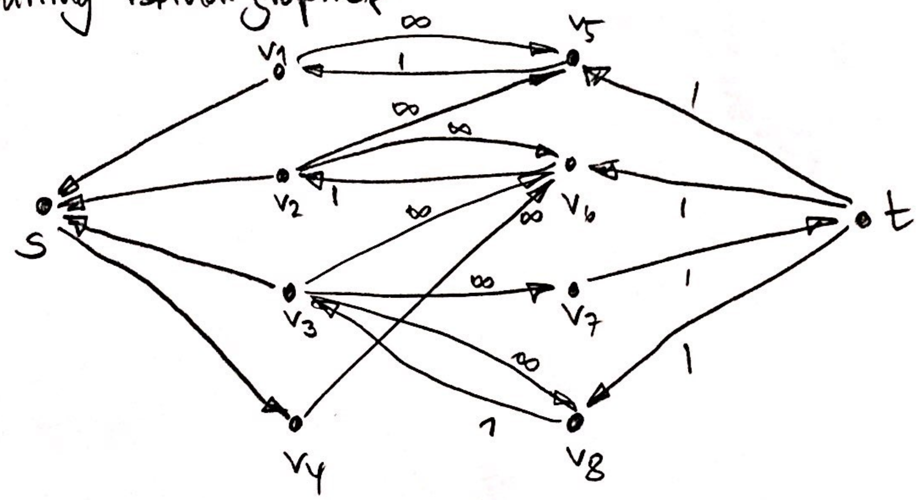


(a) (G_1', c, s, t)



(b) Flow of value 3

resulting residual graph G_f':



There exists no s-t-path in G_f' → flow is optimal

⇒ matching in G: