Q5 Thurs ≡ exam 3 solutions ⊕ final review

Final ≡ exams 1 to 3   w/ only enough problems
to get done in 2 hrs.
(most likely 5 prbs / exam)

2.3 (7a)
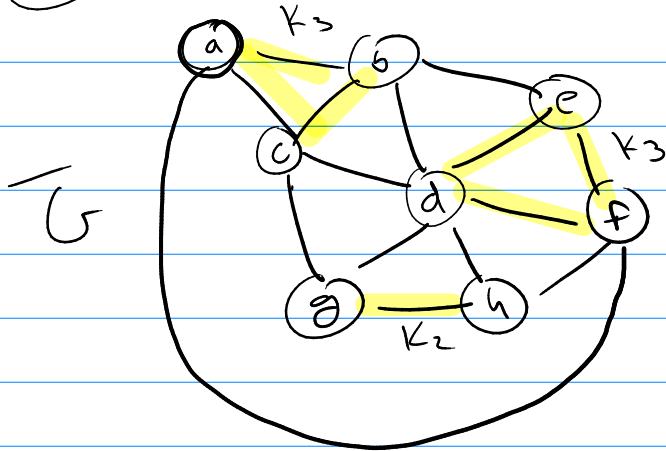
Set B color
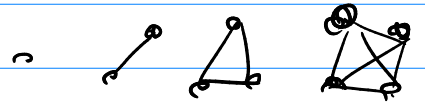Mutually Non-adj. vertices in G

K_3

$\overline{G}$

K_3

K_2

→ K_3, K_3, K_2

in $\overline{G}$ sets of mutually adj. vertices (complete subgraphs)

$K_n - G$

$K_1, K_2, K_3, K_4, .$

4 coins

a vs b

n odd   1 good coin = g

3-ary

(1ℓ)(1h)(2ℓ)(2h)(3ℓ)(3h)(4ℓ)(4h), (ok) ← outcomes
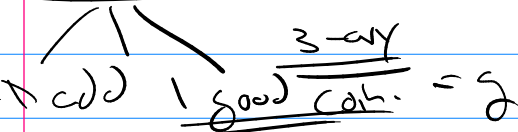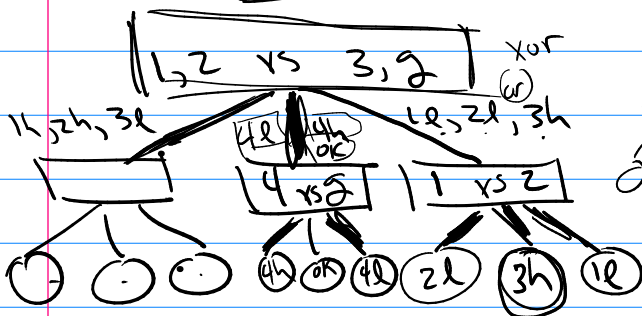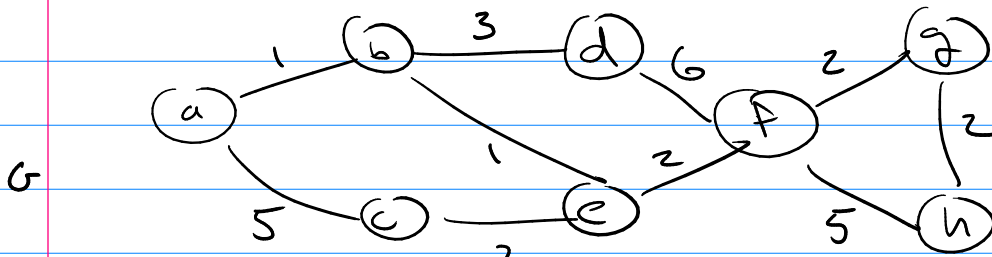
$h \geq \lceil \log_3 9 \rceil = 2$

1,2 vs 3,g   xor (w)

1ℓ,2h,3ℓ   4ℓ/4h/ok   1ℓ,2ℓ,3h

4 vs g   1 vs 2

4h ok 4ℓ 2ℓ 3h 1ℓ

Networks ≡ Weigthed graph

G

$$M_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} a\,b\ c\,d\ e\,f\ g\,h \\ \left[\begin{array}{cccccccc} 0 & 1 & 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 3 & 1 & 0 & 0 & 0 \\ & & & & \boxed{e\,|\,2} \end{array} \right] \end{array}$$

tasks? ① spanning tree

② find minimal paths from a vertex (root) to all other vertices (Dijkstra's Algorithm)

cost of tree : 17

dbei:4   db:3   d:0   dfg:8   df:6

dbec:6   dbe:4   afgh:10

Know   unknown
d,b,a,e   c, f,g,h

we get :-
① from root to every vertex a path of least cost

② get a spanning tree that minimizes cost from root.
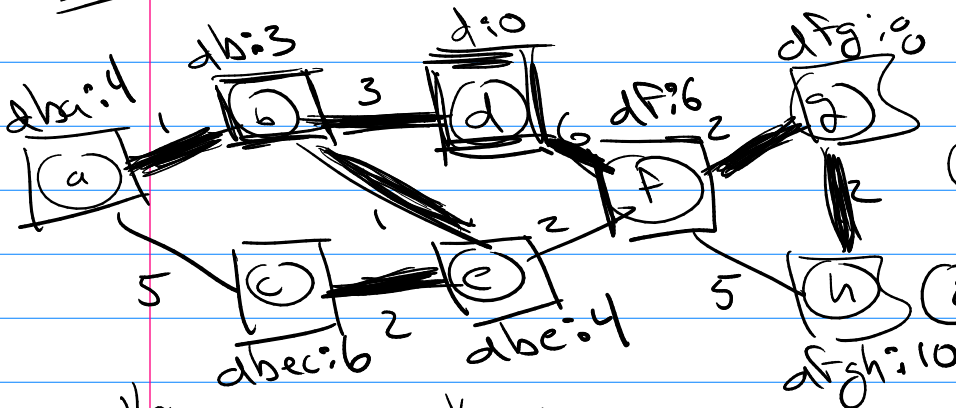
Dijkstra's Algorithm
① pick root (1x) d

② go from known to unknown in path from root of least cost.

| Vertex	path	cost
→ d	d	0
. b	db	③
. a	dba	④
. e	dbe	④
. c	dbec	6

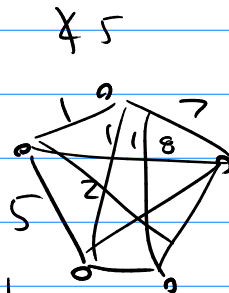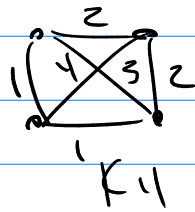Costs

1, 2, 3, 5, 6

Cost: 13

[or]
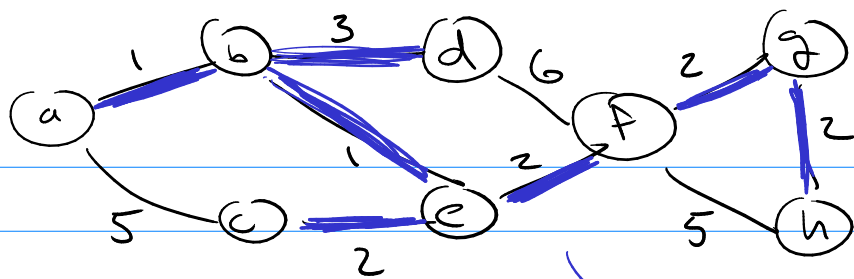
Kruskal's spanning tree
from G select |V|-1 edges such that
in order of cheapest up choose all edges
that do not form a circuit

Prim's Spanning Tree
from G select the least cost edge first.
then from the vertices of selected edges pick
next vertices connected by least cost edge
that do not form simple circuit.
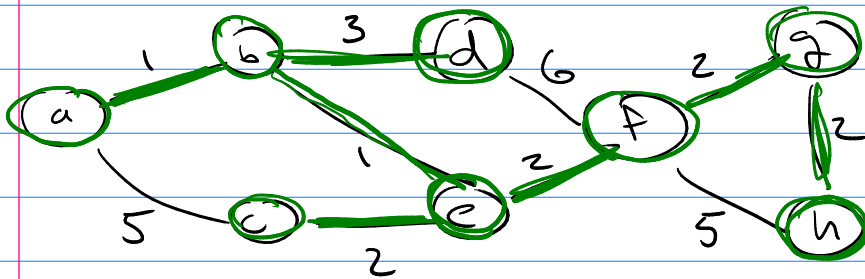
Traveling Salesman Problem ; K_n Network

K_2 2

K_3 1 1 3

K_4 2 1 4 3 2 1

K_5 1 9 7 1 8 5 2 6

Note: each K_n has $(n-1)!$ hamilton circuits.

traveling Salesman problem is ..

of $(n-1)!$ hamilton circuits whose
cost is sum of its edges ..

[Problem] find circuit of least cost.

Section 3.3 has an approximate algorithm

the real ans. algorithm is a NP Problem
countably Infinite

have an effective
algorithm
to solve →

Solvable   Unsolvable $\cup$ = all probs

$\underline{\underline{no}}$ effective
algorithm to
solve

↑
uncountable

NP
P

uncountably
infinite

π 3.1 --