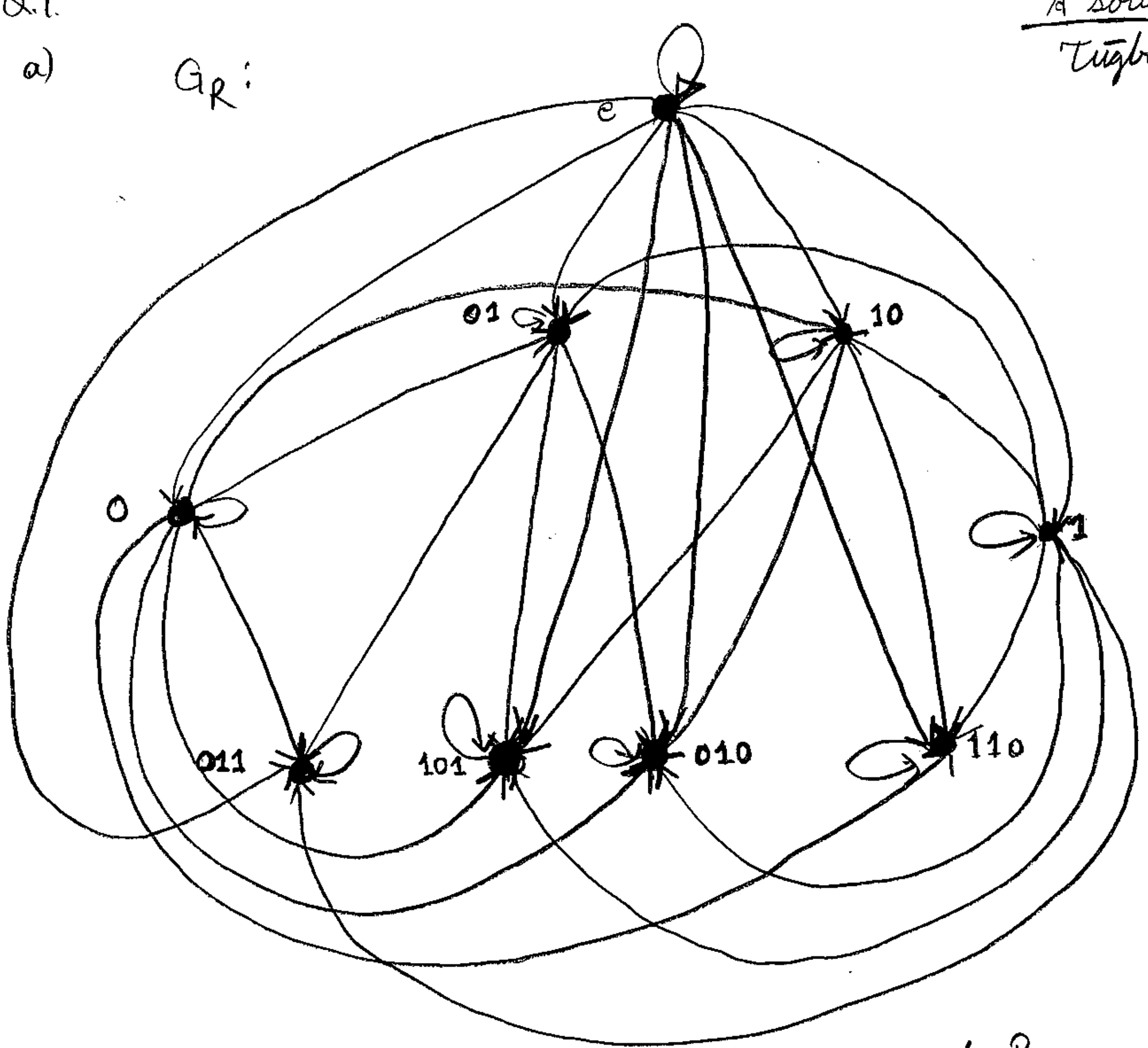
Q.1.

A solution for HW#5,
Tugberk.

a) G_R:



$G_R = (V_R, E_R)$ is a <u>digraph</u> to represent R

where $V_R = \{e, 0, 1, 01, 10, 010, 011, 101, 110\}$ and

$E_R = \{ (e,e), (e,0), (e,1), (e,01), (e,10), (e,010), (e,011), (e,101), (e,110),$
$(0,0), (0,01), (0,10), (0,010), (0,011), (0,101), (0,110),$
$(1,1), (1,01), (1,10), (1,010), (1,011), (1,101), (1,110),$
$(01,01), (01,010), (01,011), (01,101),$
$(10,10), (10,010), (10,101), (10,110),$
$(010,010),$
$(011,011),$
$(101,101),$
$(110,110) \}$.

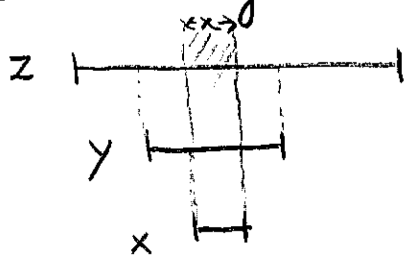
b) (S,R) is a poset iff R on S is a partial-order relation.

R is reflexive since for all $x \in S$, $(x,x) \in R$ as x is a substring of itself by definition.

R is antisymmetric as for all $x,y \in S$, if $(x,y) \in R$ and $(y,x) \in R$ then $x=y$. Using G_R in (a), you may deduce that R is antisymmetric since apart from the self-loops due to reflexivity, no two edges of the form (v_1, v_2) and (v_2, v_1) s.t. $v_1, v_2 \in V_R$ occur in E_R at the same time.

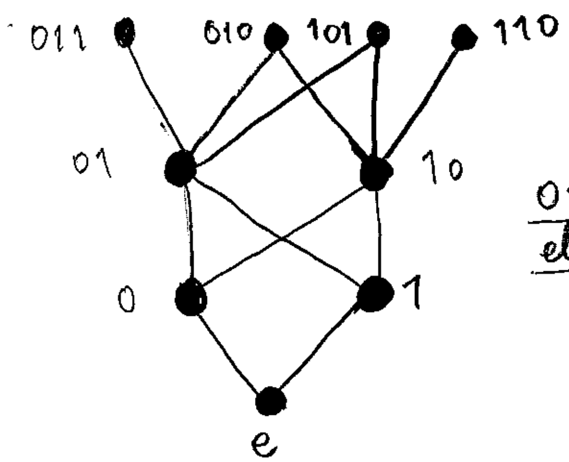
: This case is not observed for any $(v_1, v_2) \in V_R$.

R is transitive since for all $x,y,z \in S$ if $(x,y) \in R$ and $(y,z) \in R$, it means that x is a substring of y and y is a substring of z, then y is contained within z, and within where y occurs in z, x occurs within y by definition. Consequently, x is a substring of z as well, making $(x,z) \in R$; i.e. pictorially:



c) R is not a total order since neither $(0,1)$ nor $(1,0)$ is a member of R even though $0,1 \in S$; not every pair of elements of (S,R) is comparable.

d) Hasse diagram for (S,R) : eliminate self-loops, directed arcs due to transivity, and directions on arcs via introducing an order for undirected arcs (bottom-up).



e is the only minimal element.

010, 011, 101, 110 are the maximal elements.

e) Least upper bounds of 0 and 1 are 01 and 10 but they are not comparable, hence no unique LUB for every pair of objects exist, and (S, R) does not constitute a lattice.

Q.2.

a)

vertex	adjacent vertices
a	a, b, d
b	c, d
c	b
d	c
e	b, f
f	f, b, e
g	c, f

b)

to

$$M_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

from

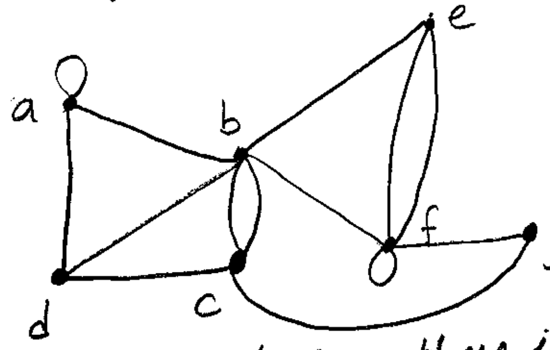
c)

vertex, v	$\deg^+(v)$	$\deg^-(v)$
a	1	3
b	4	2
c	3	1
d	2	1
e	1	2
f	3	3
g	0	2

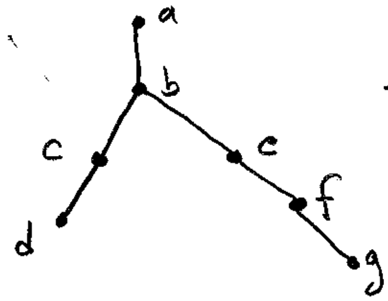
d) a, a, b, d, c.
b, c, b, d, c.
f, e, f, f, b.
g, f, b, d, c.
b, d, c, b, c.
f, f, e, b, c.

e) b, d, c, b.
c, b, d, c.
d, c, b, d.
f, f, e, f.
f, e, f, f.
e, f, f, e.

f) Underlying undirected graph G' of G is as follows:



G' is connected since there is a path between every vertex pair, i.e. when you perform DFS on G' it yields a tree:

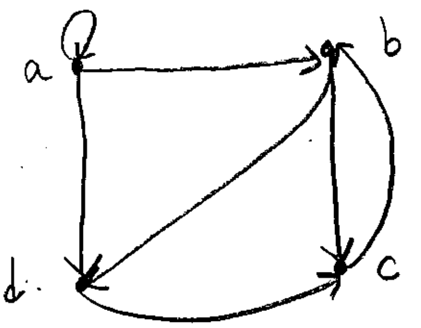


and every tree is connected by def$^{\underline{n}}$.

g) SCC's of G are $\{a\}, \underbrace{\{b, c, d\}}, \{e, f\}, \{g\}$.

there is a path between every vertex pair & is maximal.

h) H is as follows:



Adjacency matrix representation of H is

$$M_H = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

(corresponding to upper-left portion of M_G in b)

Different paths of length 3 for every distinct vertex pair of H is

$$\underbrace{M_H^3} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 3 & 3 & 2 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

Q.3.

vertex	a	b	c	d	e	f	g	h	i	j	k
deg	2	6	4	4	6	6	4	4	4	6	2

a. Since all vertices have even degree, G does <u>not</u> have a Euler path. (Usually EP-EC ; HP-HC are defined exclusively; i.e. an Euler path starts and ends at different vertices.)

b. Since all vertices have even degree, G has a Euler circuit.
(Enough to be deemed as correct answer.)
Additionally, let us show it using the algorithm in the textbook: (Fleury's algorithm)
→ Form a circuit choosing edges arbitrarily & remove those edges and proceed iteratively till all edges are removed:

c, f, b, b, e, h, h, i, c

initial circuit

append:

b, e, a, b   e, f, i, h, e   c, d, g, k, j, j, c   g, d, j, g

edges are undirected, so both paths of the form $x \rightsquigarrow y$; $y \rightsquigarrow x$ are possible when $x, y \in V_G$ and $(x,y) \in E_G$ or $(y,x) \in E_G$

c. Yes; (b, a, e, h, i, f, c, j, k, g, d) is a Hamiltonian path on G.

d. No, G does not contain a Hamiltonian circuit because c is an articulation point in G, i.e. removal of c disconnects G and so
→ for all vertices except for c when they constitute the beginning and end vertices of a HC c has to be visited at least twice to conclude the circuit.
→ for a HC beginning at c either a, b, e, f, h, i or d, g, k, j can be appended to the path via next vertices chosen as f, i, d, or j and c has to be visited again before proceeding with the other subgraph.

Q.4.

a) $K_{m,n}$ has $(m+n)$ vertices and $(m.n)$ edges

b) Let $V_L = \{a_1, a_2, \dots, a_m\}$ and $V_R = \{b_1, b_2, \dots, b_n\}$ be two disjoint sets of vertices s.t. $V_L \cup V_R = V$ is the set of vertices of $K_{m,n}$. Moreover, it is given, m is an odd number while n is even. Assume that $K_{m,n}$ has a HC. Its HC has the form $a_1', b_1', \dots a_k', b_k', a_1'$ for $K_{m,n}$ s.t. each $a_i' \in V_L$ and $b_i' \in V_R$ with $a_i' \neq a_j'$ for $i \neq j$. Then, except for a_1'; $b_1', \dots, b_k'$ and $a_2', \dots, a_k'$ is listed once and $a_1', \dots a_k', b_1', \dots, b_k'$ covers every vertex in $K_{m,n}$ so that $|V| = 2k$ for some $k \in \mathbb{N}$. In this case, the number of vertices must be an even number. However, in our $K_{m,n}$, m is odd, and n is even, making $(m+n) = |V|$ an odd number. The other form of the HC would be $b_1', a_1', \dots, b_k', a_k', b_1'$ and would similarly require $(m+n)$ to be an even number. Since $m+n$ is odd, we have reached a contradiction, and consequently K_{m+n} with m odd, n even does not have a HC.

Q.5. a)

visited	iteration	d_s	π_s	d_u	π_u	d_v	π_v	d_w	π_w	d_x	π_x	d_y	π_y	d_z	π_z	d_t	π_t
$\emptyset$	0	0	–	∞	–	∞	–	∞	–	∞	–	∞	–	∞	–	∞	–
s	1	0	–	4	s	5	s	3	s	∞	–	∞	–	∞	–	∞	–
w	2	0	–	4	s	5	s	3	s	11	w	∞	–	15	w	∞	–
u	3	0	–	4	s	5	s	3	s	11	w	15	u	15	w	∞	–
v	4	0	–	4	s	5	s	3	s	7	v	11	v	15	w	∞	–
x	5	0	–	4	s	5	s	3	s	7	v	8	x	13	x	∞	–
y	6	0	–	4	s	5	s	3	s	7	v	8	x	12	y	17	y
z	7	0	–	4	s	5	s	3	s	7	v	8	x	12	y	15	z
t	8	0	–	4	s	5	s	3	s	7	v	8	x	12	y	15	z

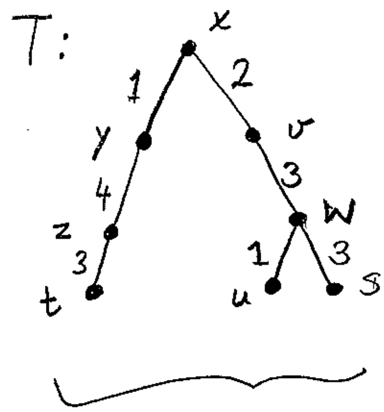
notation d_v : shortest distance to vertex v from s

π_v : the vertex visited before v on the shortest path from s to v.

Thus, the shortest path from s to t is (s, v, x, y, z, t).

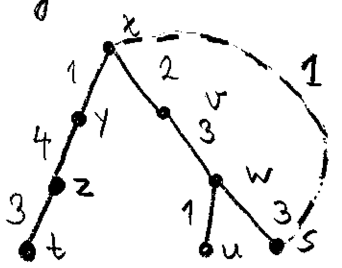
b)

choice	edge	weight
1	$\{x, y\}$	1
2	$\{x, v\}$	2
3	$\{v, w\}$	3
4	$\{w, u\}$	1
5	$\{w, s\}$	3
6	$\{y, z\}$	4
7	$\{z, t\}$	3

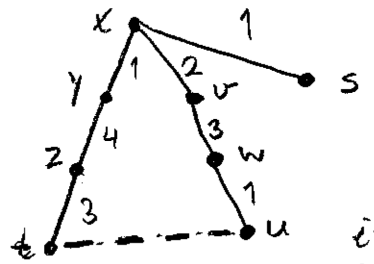


c) Addition of a new edge creates one cycle in T. If the weight of the added edge is less than the maximum weight of other edges in the cycle, then remove the edge with max weight from T and add the new edge to the tree. Otherwise, keep the tree as is.

Initially, T is:

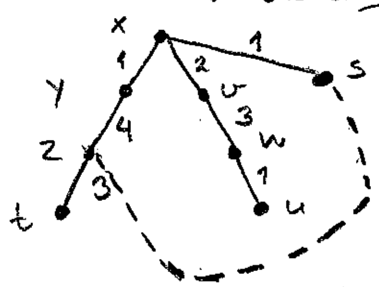


i
(s, x, 1)
added
either remove (v, w, 3) or (w, s, 3)
add (s, x, 1)



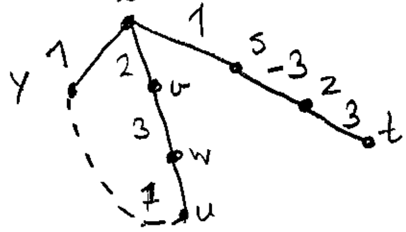
is still a MST

ii
(t, u, 6) added
6 < 4, so
tree does not change

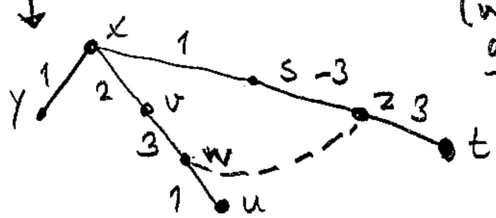


iii
(s, z, -3)
added

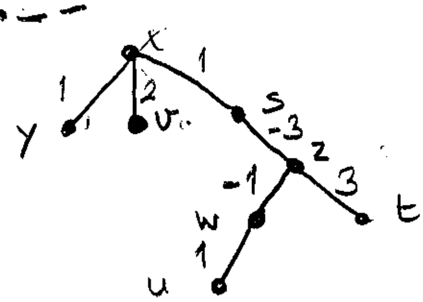
remove (y, z, 4)
add (s, z, -3)



(u, y, 3) iv
added,
3 < 3 doesn't hold, so
tree does not change

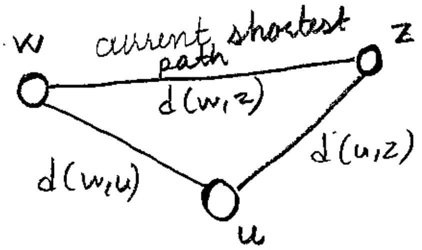


(w, z, -1) v
added
-1 < 3
remove (v, w, 3)
add (w, z, -1)



final MST for the updated G.

d) No, it does not work the same manner in c)
because Dijkstra's algorithm relies on triangle inequality governing the path update rule as follows:

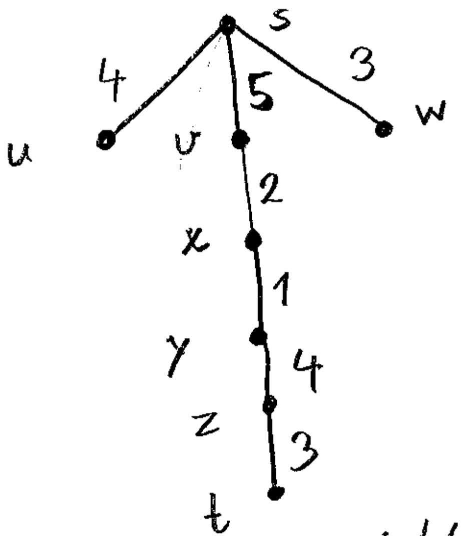


Modify shortest path from w to z as w, u, z iff

$$\boxed{d(w,u) + d(u,z) \leq d(w,z)}$$

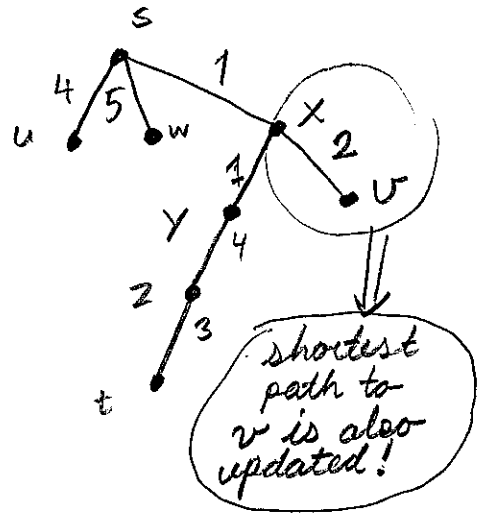
For comparison, MST algorithms use only edge weights w/o prior info.

e.g. Shortest path tree generated in a) is:



(s, x, 1) added

tree is changed as:



Secondly, negative weight edges might tempt a greedy algorithm such as Dijkstra's to make short-sighted choices and fail to produce correct answers. Consider after (s, z, -3) and (w, z, -1) are added to G, the shortest path from s to s is no longer through itself at 0 cost, but via s, z, w, s with a cost of -1, yet Dijkstra's algorithm fails to discover that as initially s is explored and removed from the closed list.

(Two-sentence explanation is OK.)

Q.6.

a) 13 vertices and 12 edges. Height is 4, the length of the path p, r, u, y, n.
Always |V| and |V|-1.

b) w, s, m, t, q, x, n, y, u, z, v, r, p.

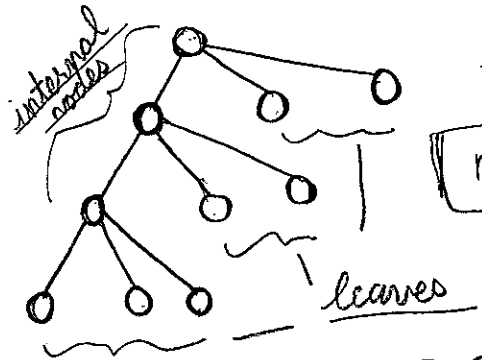
c) s, w, q, m, t, p, x, u, n, y, r, v, z.

d) p, q, s, w, t, m, r, u, x, y, n, v, z.

e) No, it is not a full binary tree. We have $m = 2$, and we have 8 internal (non-leaf) nodes so, $i = 8$. According to the theorem, we must have had $n = mi + 1 = 17$ vertices, yet we have 13 vertices.

f) $x : 41 > p : 42$ should hold, yet it doesn't.
$n : 61 > u : 63$ should hold, yet it doesn't.
Thus, T is not a BST.

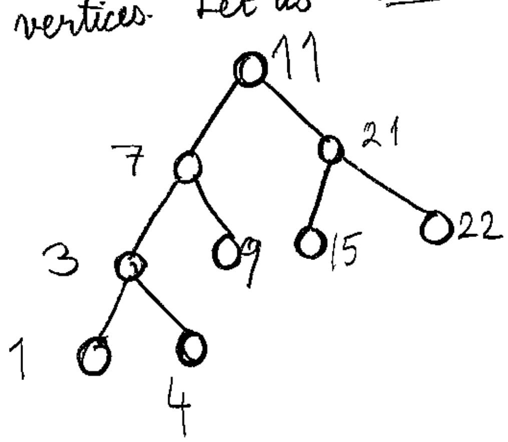
g)



$m = 3$
$i = 3$, to make height 3 as required.

$$\boxed{n = mi + 1 = 3 \cdot 3 + 1 = 10.}$$

h) $K = \{9, 3, 11, 15, 1, 7, 22, 21, 4\}$.
$|K| = 9$ and minimum height tree is a complete tree with 9 vertices. Let us construct the BST level-by-level:



⇒ has height 3
Sort K as (1, 3, 4, 7, 9, 11, 15, 21, 22)
carry out inorder traversal on the BST and as a vertex is visited, put the first constituent of the sorted tuple there and diminish the tuple via removing the put value.

Other BST's are OK so long as
i) they are BST's. ✓
ii) their height is 3. ✓

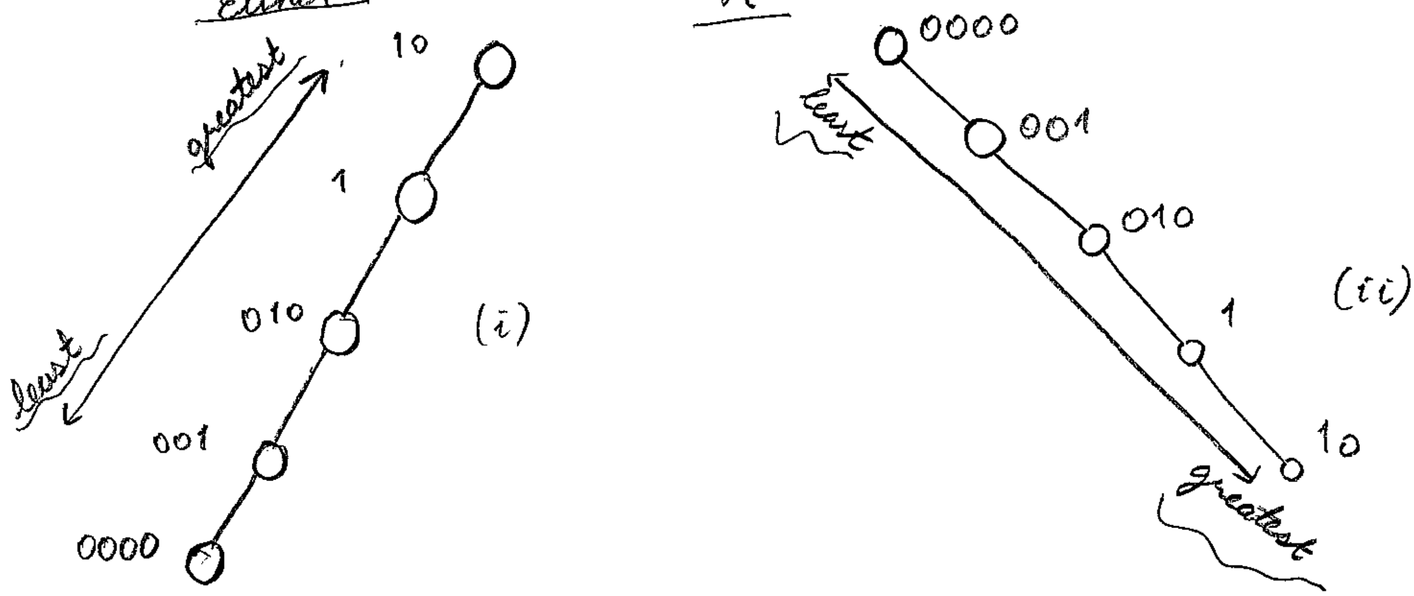
i) To find 2, vertices 11, 7, 3 and 1 are probed.
To find 22, " 11, 21, and 22 are probed.

j) Let < denote the lexicographic comparison relation and K = {001, 1, 10, 010, 0000}
then ordering K wrt <, we get the tuple:

$\boxed{(0000, 001, 010, 1, 10)}$.

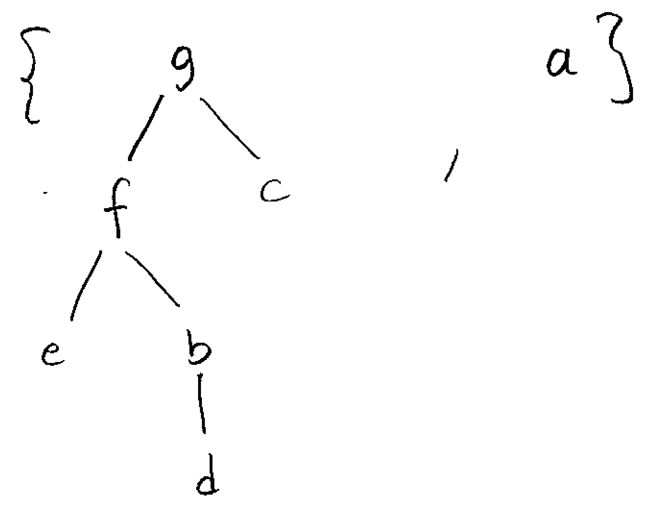
A BST with max height should have linear structure and height 4:

Either ... or



k) on (i), to find 001;
10, 1, 010, 001 are probed;
, to find 011;
10, 1, 010 are probed.

on (ii), to find 001;
0000, 001 are probed;
, to find 011;
0000, 001, 010, 1 are probed.

l) 

is the spanning forest with two member trees as indicated.