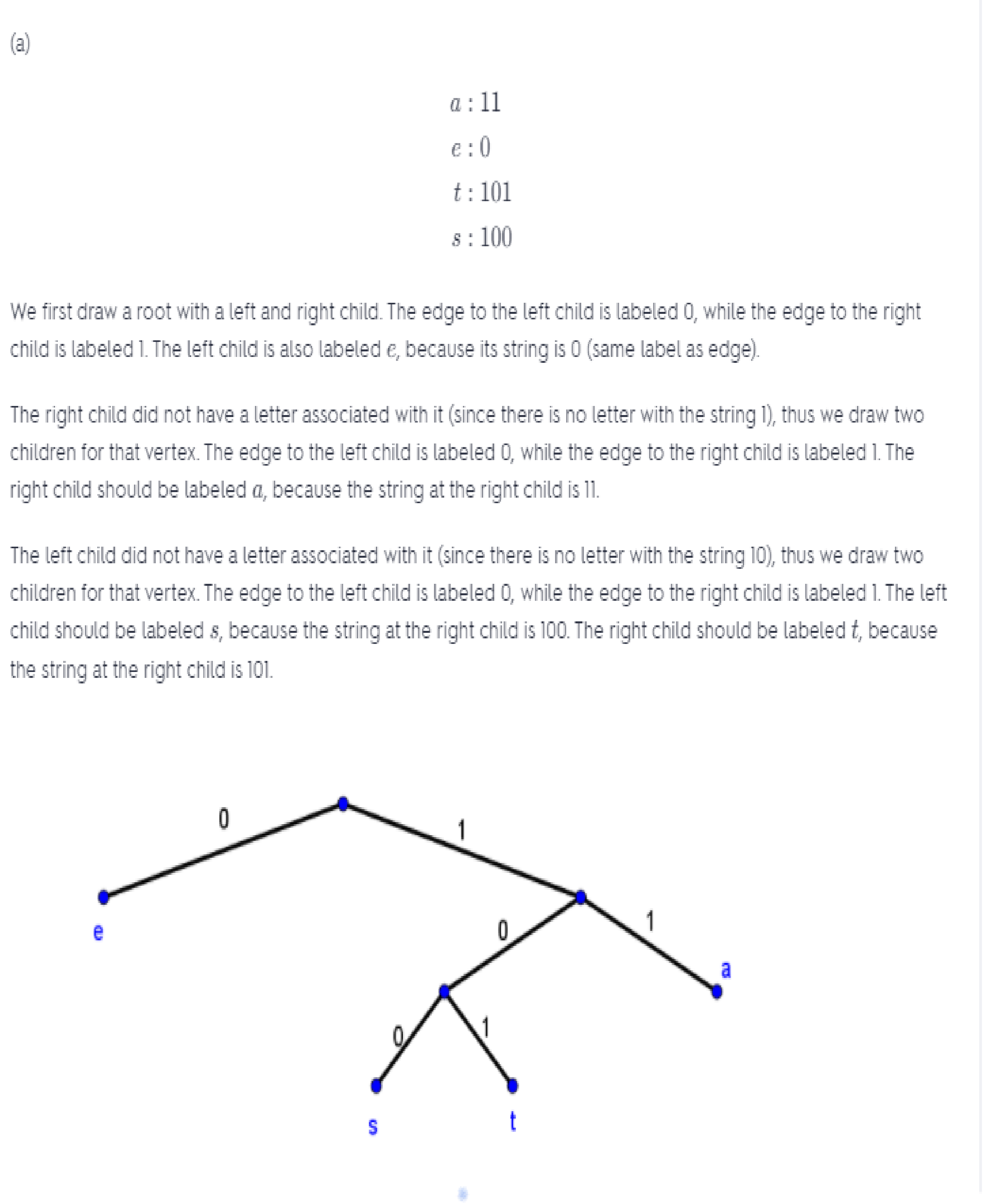
11.2(20)



(a)

$$a : 11$$
$$e : 0$$
$$t : 101$$
$$s : 100$$

We first draw a root with a left and right child. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The left child is also labeled e, because its string is 0 (same label as edge).

The right child did not have a letter associated with it (since there is no letter with the string 1), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child should be labeled a, because the string at the right child is 11.

The left child did not have a letter associated with it (since there is no letter with the string 10), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The left child should be labeled s, because the string at the right child is 100. The right child should be labeled t, because the string at the right child is 101.

(b)

$$a : 1$$
$$e : 01$$
$$t : 001$$
$$s : 0001$$
$$n : 00001$$

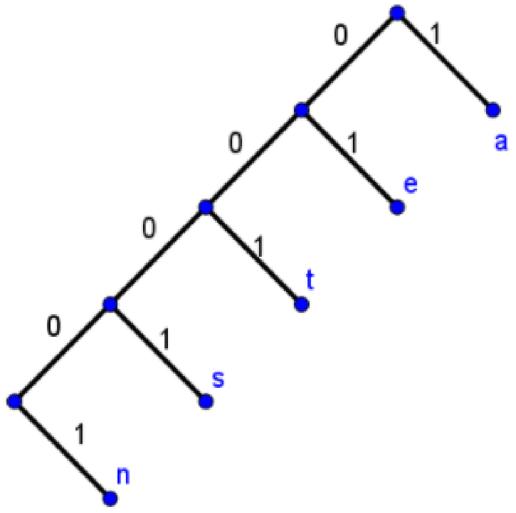
We first draw a root with a left and right child. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child is also labeled a, because its string is 1 (same label as edge).

The left child did not have a letter associated with it (since there is no letter with the string 0), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child should be labeled e, because the string at the right child is 01.

The left child did not have a letter associated with it (since there is no letter with the string 00), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child should be labeled t, because the string at the right child is 001.

The left child did not have a letter associated with it (since there is no letter with the string 000), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child should be labeled s, because the string at the right child is 0001.

The left child did not have a letter associated with it (since there is no letter with the string 0000), thus we draw one right child for that vertex (as the only string remaining is 00001). The edge to the right child is labeled 1. The right child should be labeled n, because the string at the right child is 00001.



(c)

$$\begin{aligned} a &: 1010 \\ e &: 0 \\ t &: 11 \\ s &: 1011 \\ n &: 1001 \\ i &: 10001 \end{aligned}$$

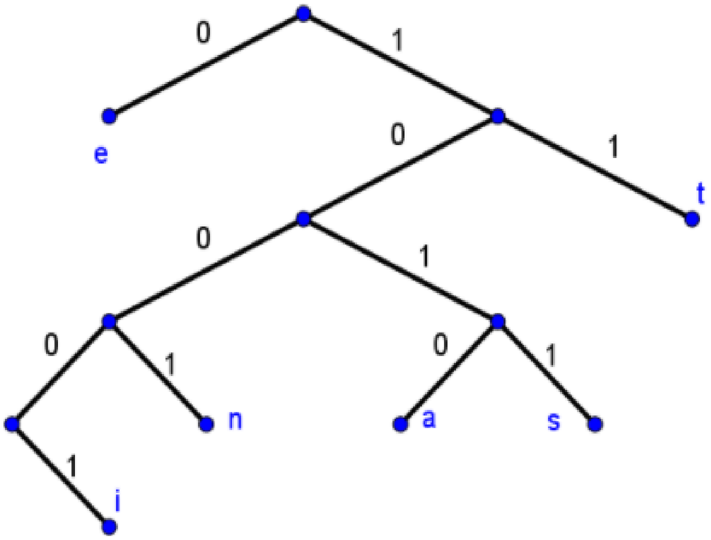
We first draw a root with a left and right child. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The left child is also labeled e, because its string is 0 (same label as edge).

The right child did not have a letter associated with it (since there is no letter with the string 1), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. The right child should be labeled t, because the string at the right child is 11.

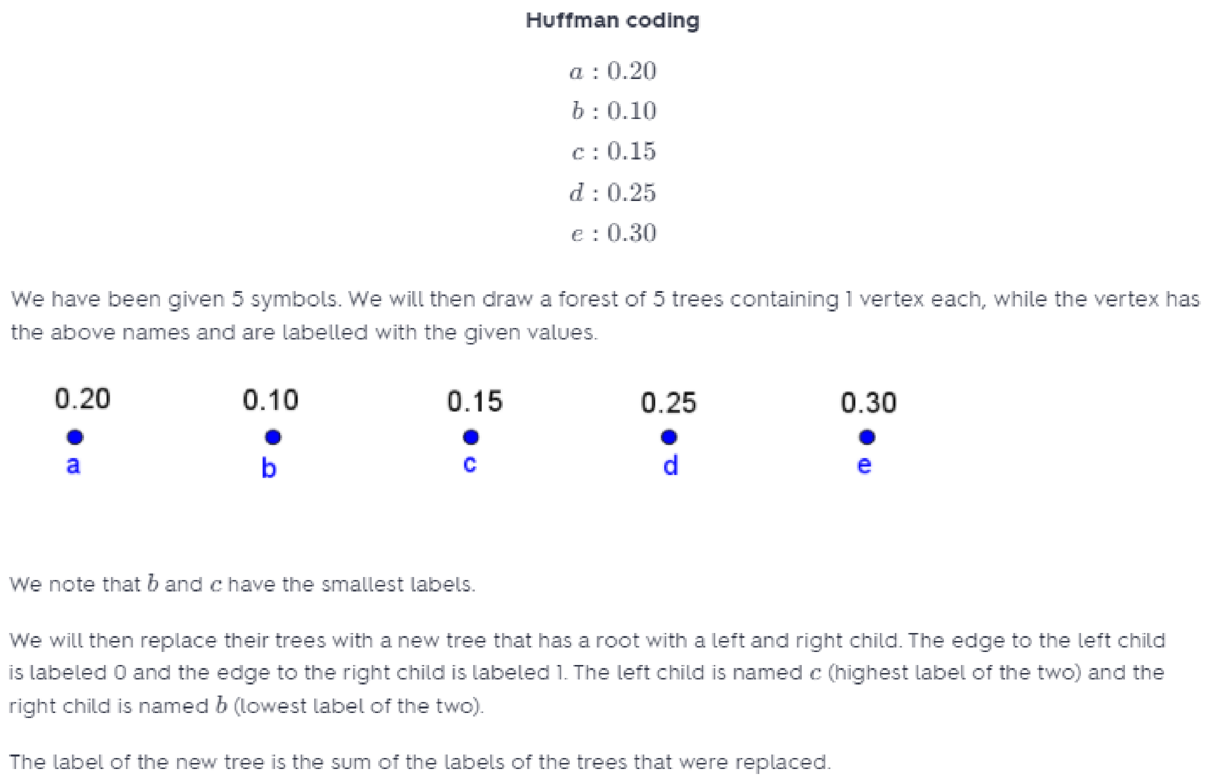
The left child did not have a letter associated with it (since there is no letter with the string 10), thus we draw two children for that vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. Neither vertex should be labeled, as there are no string with 3 digits.

Since neither child had a letter associated with it, we draw two children for each vertex. The edge to the left child is labeled 0, while the edge to the right child is labeled 1. Three of the four children should be labeled: a for strings 1010, s for string 1011 and n for string 1001.

We draw a right child for the remaining child that did not have a letter associated with it (since only the string 10001 is left and the 1 should be a right child). The edge to the right child is labeled 1. The right child should be labeled i, because the string at the right child is 10001.



11.2(23)



Huffman coding

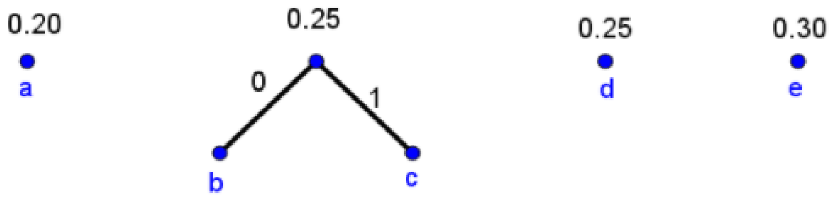
$$a : 0.20$$
$$b : 0.10$$
$$c : 0.15$$
$$d : 0.25$$
$$e : 0.30$$

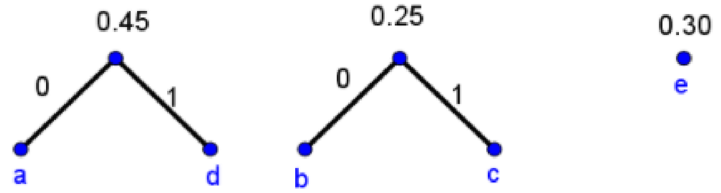
We have been given 5 symbols. We will then draw a forest of 5 trees containing 1 vertex each, while the vertex has the above names and are labelled with the given values.

We note that b and c have the smallest labels.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named c (highest label of the two) and the right child is named b (lowest label of the two).

The label of the new tree is the sum of the labels of the trees that were replaced.

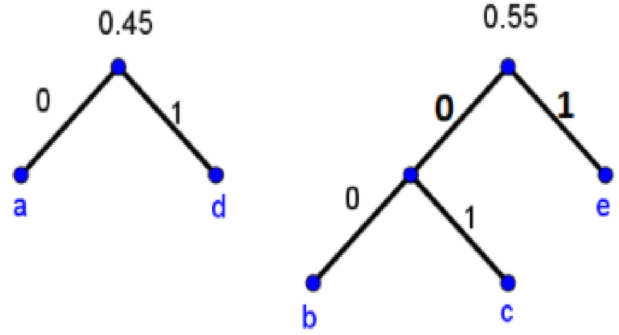




We then note that the trees with the smallest labels are $b+c$ and e.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named e (highest label of the two) and the right child is the tree $b+c$ (lowest label of the two).

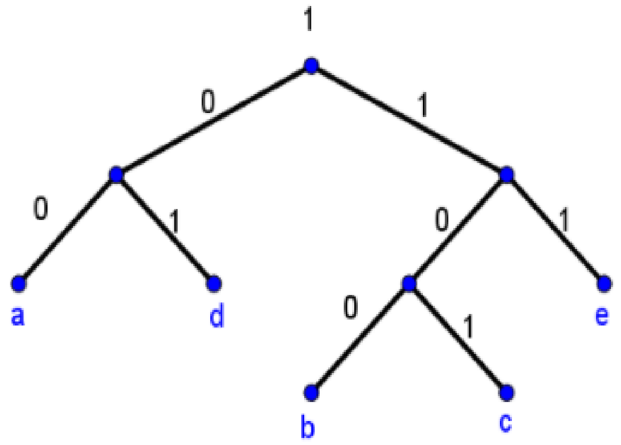
The label of the new tree is the sum of the labels of the trees that were replaced.

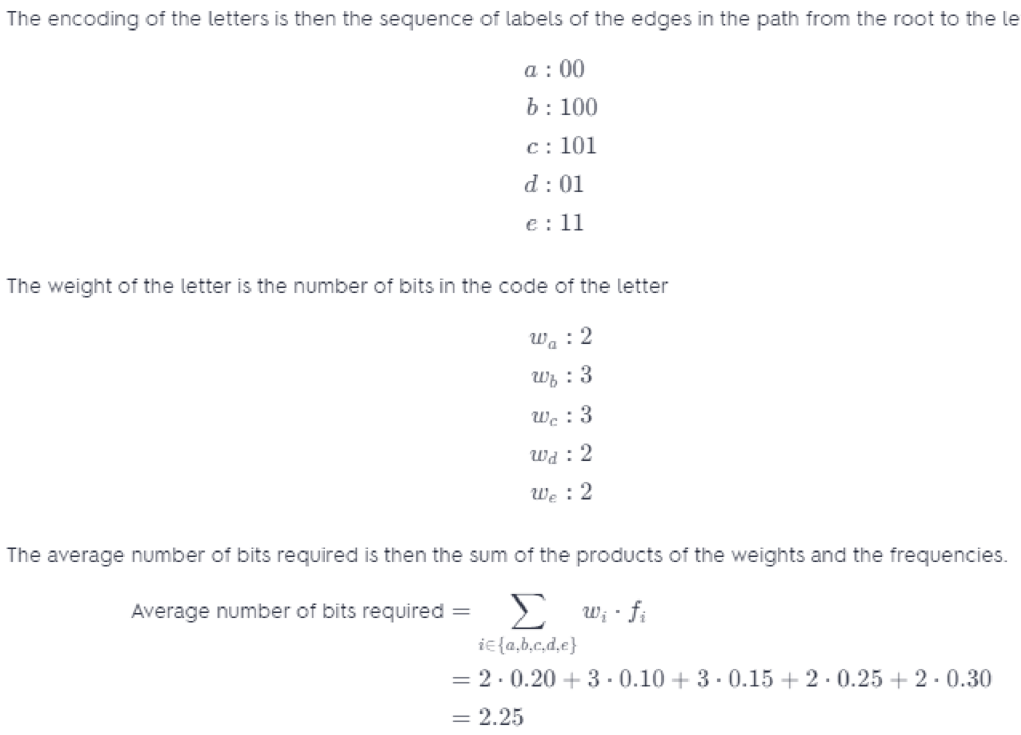


We then note that the trees with the smallest labels are $b+c+e$ and $a+d$.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named $b+c+e$ (highest label of the two) and the right child is the tree $a+d$ (lowest label of the two).

The label of the new tree is the sum of the labels of the trees that were replaced.





The encoding of the letters is then the sequence of labels of the edges in the path from the root to the letter.

$$a : 00$$
$$b : 100$$
$$c : 101$$
$$d : 01$$
$$e : 11$$

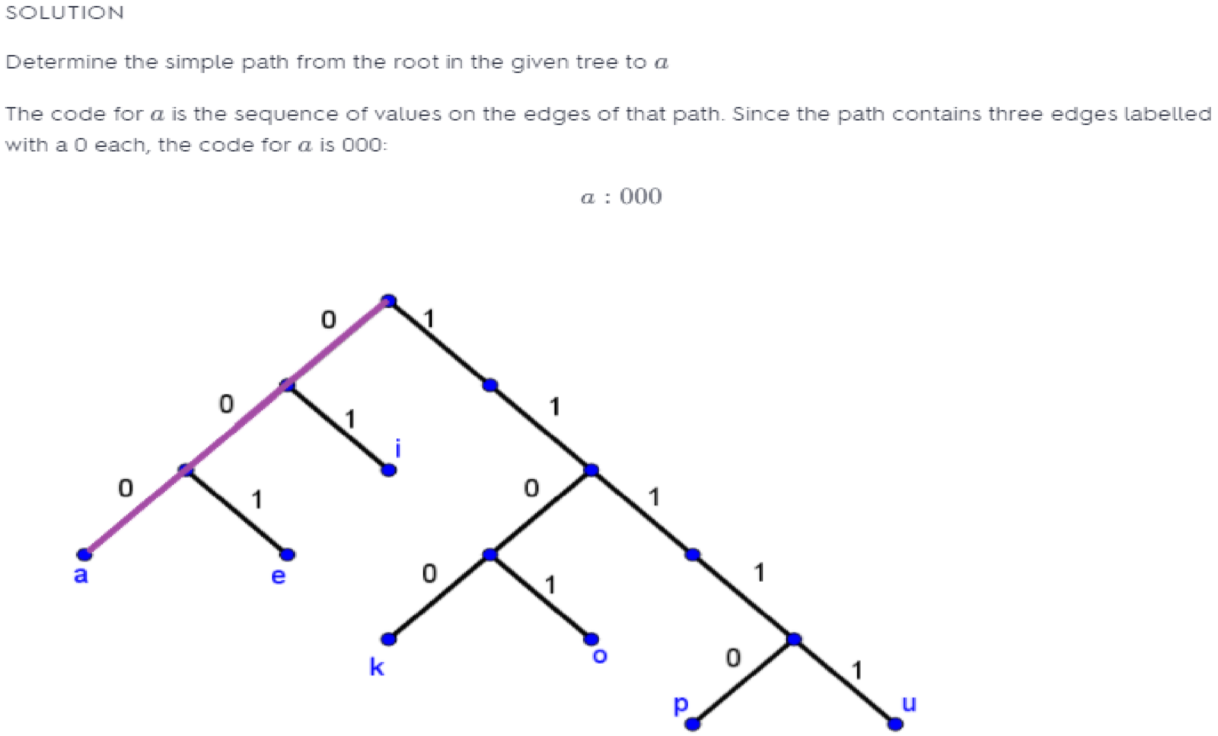
The weight of the letter is the number of bits in the code of the letter

$$w_a : 2$$
$$w_b : 3$$
$$w_c : 3$$
$$w_d : 2$$
$$w_e : 2$$

The average number of bits required is then the sum of the products of the weights and the frequencies.

$$\text{Average number of bits required} = \sum_{i\in\{a,b,c,d,e\}} w_i \cdot f_i$$
$$= 2 \cdot 0.20 + 3 \cdot 0.10 + 3 \cdot 0.15 + 2 \cdot 0.25 + 2 \cdot 0.30$$
$$= 2.25$$

11.2(21)



SOLUTION

Determine the simple path from the root in the given tree to a

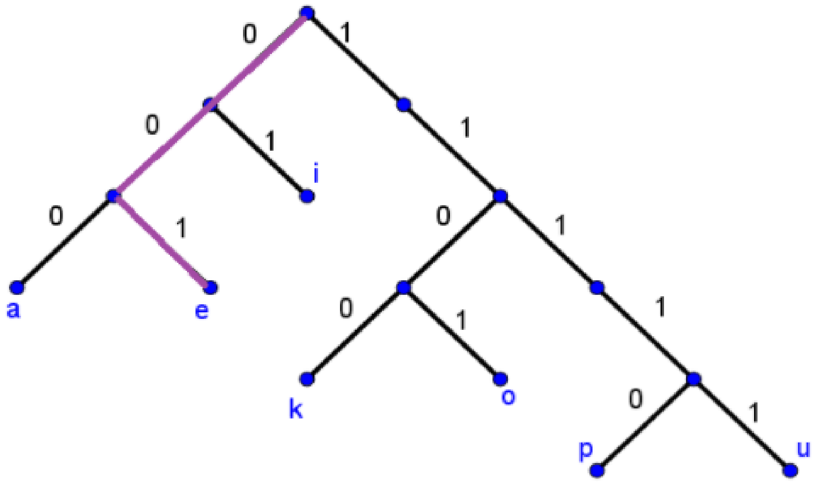
The code for a is the sequence of values on the edges of that path. Since the path contains three edges labelled with a 0 each, the code for a is 000:

$$a : 000$$

Determine the simple path from the root in the given tree to e

The code for e is the sequence of values on the edges of that path. Since the path first uses two edges labelled 0 and then an edge labelled 1, the code for e is 001:

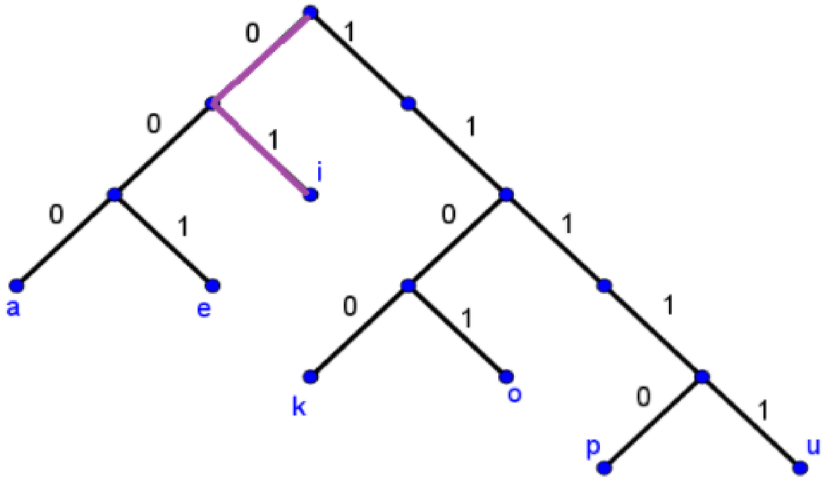
$$e : 001$$



Determine the simple path from the root in the given tree to i

The code for i is the sequence of values on the edges of that path. Since the path first uses an edge labelled 0 and then an edge labelled 1, the code for i is 01:

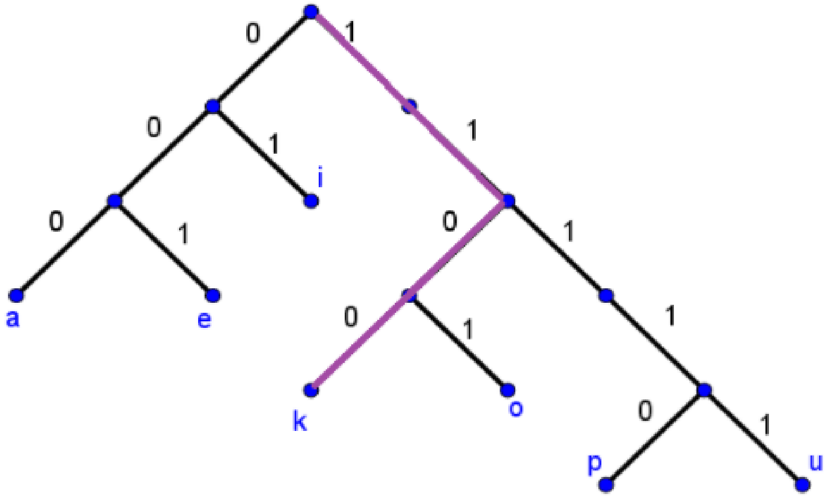
$$i : 01$$



Determine the simple path from the root in the given tree to k

The code for k is the sequence of values on the edges of that path. Since the path first uses two edges labelled 1 and then two edges labelled 0, the code for k is 1100:

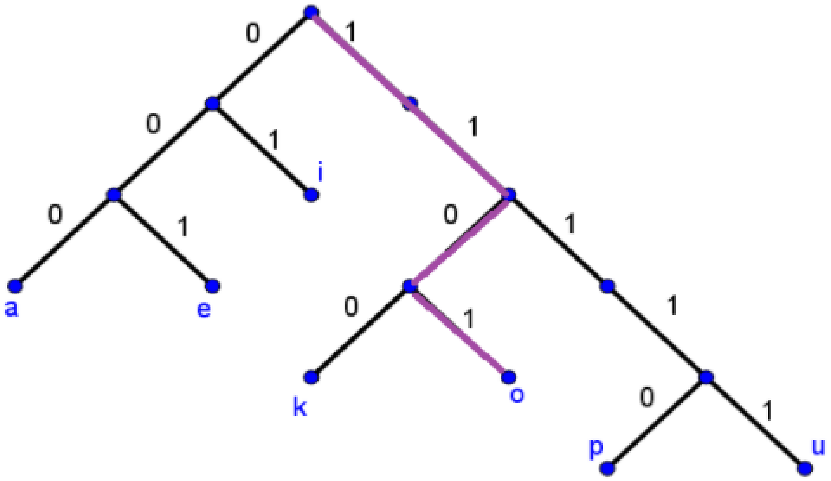
$$k : 1100$$



Determine the simple path from the root in the given tree to o

The code for o is the sequence of values on the edges of that path. Since the path first uses two edges labelled 1, then one edge labelled 0 and then one edge labelled 1, the code for o is 1101:

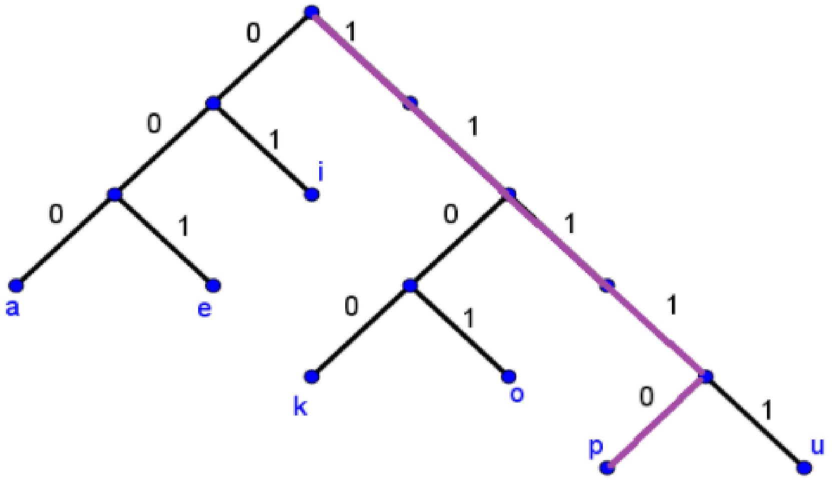
$$o : 1101$$



Determine the simple path from the root in the given tree to p

The code for p is the sequence of values on the edges of that path. Since the path first uses four edges labelled 1, then one edge labelled 0, the code for p is 11110:

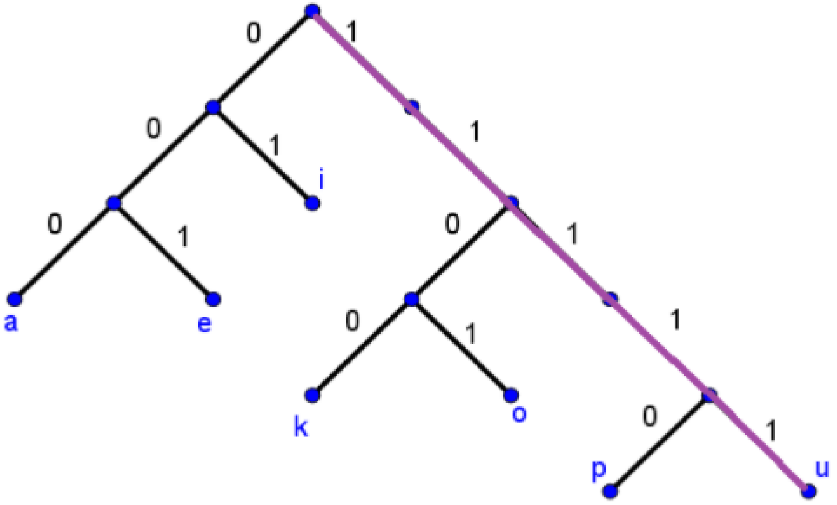
$$p : 11110$$



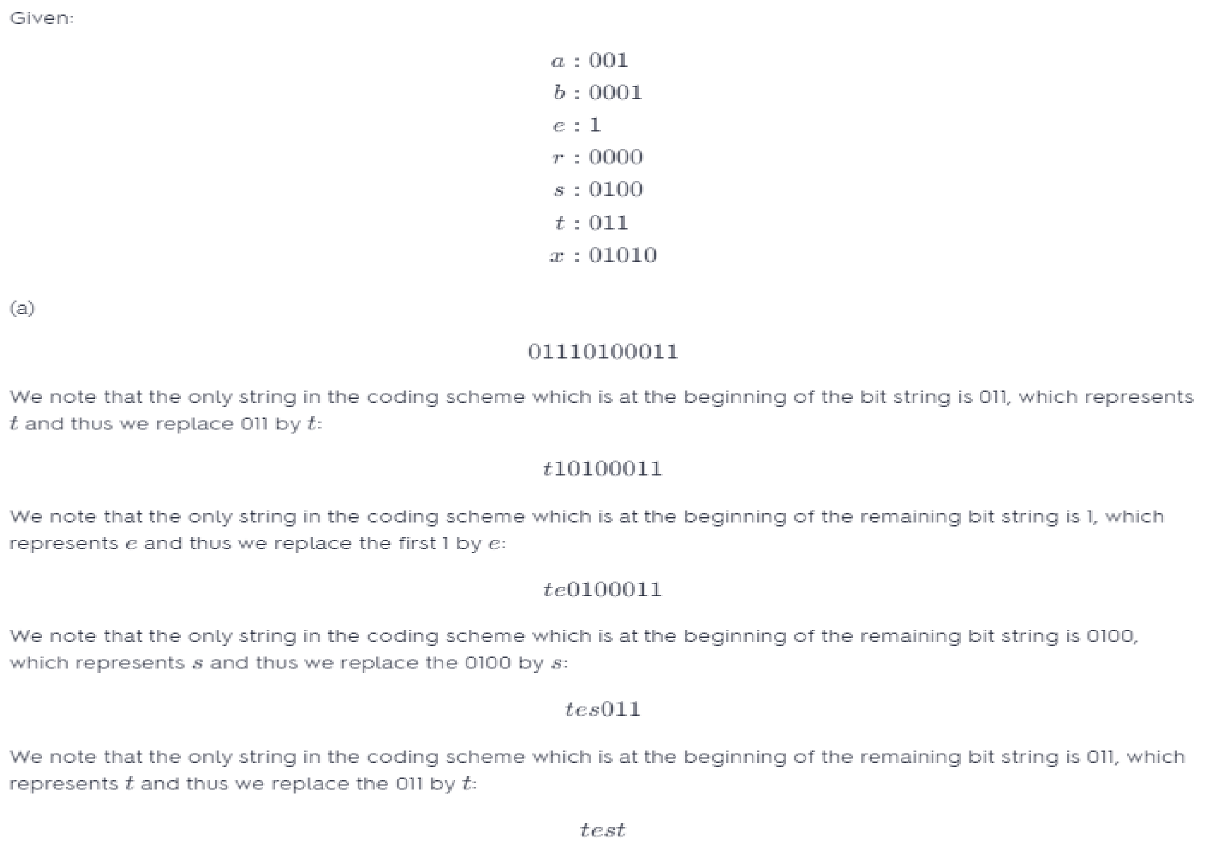
Determine the simple path from the root in the given tree to u

The code for u is the sequence of values on the edges of that path. Since the path uses five edges labelled 1 and no other edges, the code for u is 11111:

$$u : 11111$$



11.2(22)



Given:

$$\begin{aligned} a &: 001 \\ b &: 0001 \\ e &: 1 \\ r &: 0000 \\ s &: 0100 \\ t &: 011 \\ x &: 01010 \end{aligned}$$

(a)

$$01110100011$$

We note that the only string in the coding scheme which is at the beginning of the bit string is 011, which represents t and thus we replace 011 by t:

$$t10100011$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 1, which represents e and thus we replace the first 1 by e:

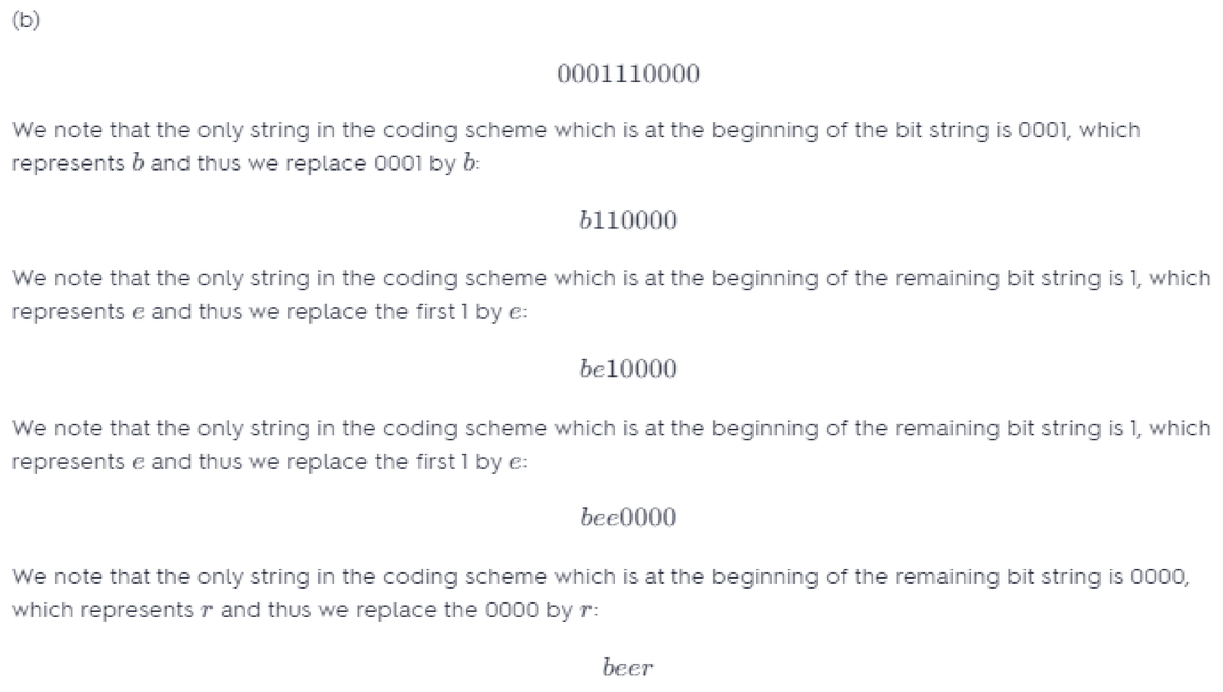
$$te0100011$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 0100, which represents s and thus we replace the 0100 by s:

$$tes011$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 011, which represents t and thus we replace the 011 by t:

$$test$$



(b)

$$0001110000$$

We note that the only string in the coding scheme which is at the beginning of the bit string is 0001, which represents b and thus we replace 0001 by b:

$$b110000$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 1, which represents e and thus we replace the first 1 by e:

$$be10000$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 1, which represents e and thus we replace the first 1 by e:

$$bee0000$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 0000, which represents r and thus we replace the 0000 by r:

$$beer$$

(c)

$$0100101010$$

We note that the only string in the coding scheme which is at the beginning of the bit string is 0100, which represents s and thus we replace the 0100 by s:

$$s101010$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 1, which represents e and thus we replace the first 1 by e:

$$se01010$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 01010, which represents r and thus we replace the 01010 by x:

$$sex$$

(d)

$$01100101010$$

We note that the only string in the coding scheme which is at the beginning of the bit string is 011, which represents t and thus we replace 011 by t:

$$t00101010$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 001, which represents a and thus we replace the 001 by a:

$$ta01010$$

We note that the only string in the coding scheme which is at the beginning of the remaining bit string is 01010, which represents r and thus we replace the 01010 by x:

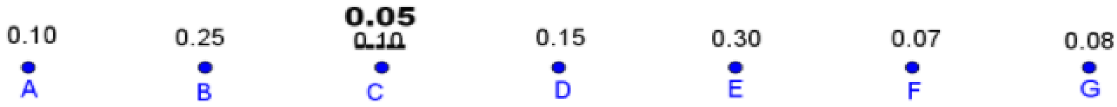
$$tax$$

11.2(22)

Huffman coding

$$A : 0.10$$
$$B : 0.25$$
$$C : 0.05$$
$$D : 0.15$$
$$E : 0.30$$
$$F : 0.07$$
$$G : 0.08$$

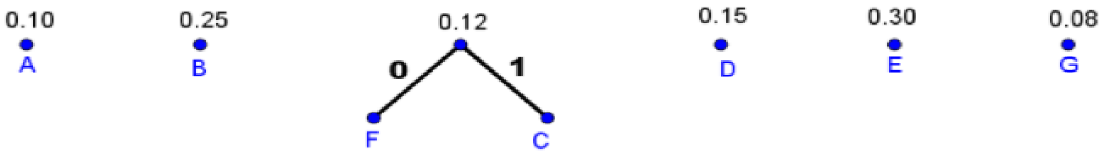
We have been given 7 symbols. We will then draw a forest of 7 trees containing 1 vertex each, while the vertex has the above names and are labelled with the given values.



We note that C and F have the smallest labels.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named F (highest label of the two) and the right child is named C (lowest label of the two).

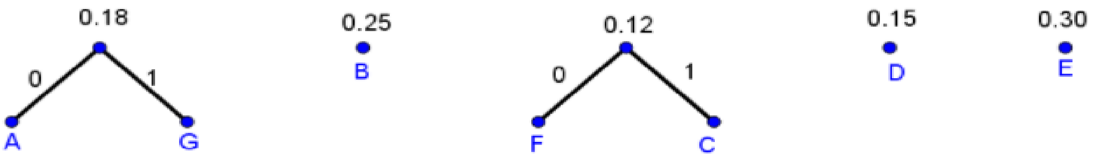
The label of the new tree is the sum of the labels of the trees that were replaced.

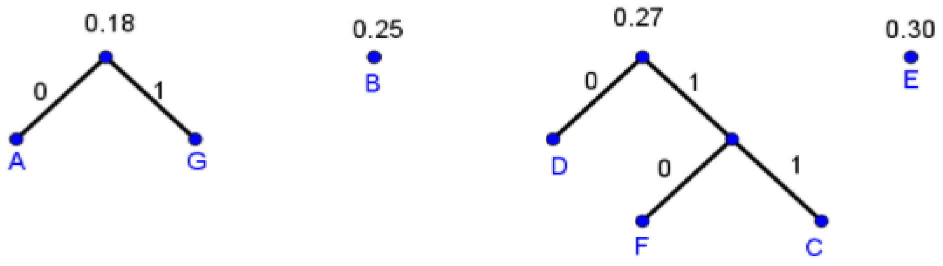


We then note that the trees with the smallest labels are A and G.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named A (highest label of the two) and the right child is named G (lowest label of the two).

The label of the new tree is the sum of the labels of the trees that were replaced.

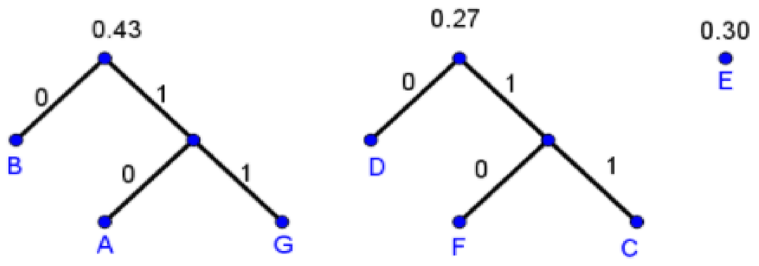




We then note that the trees with the smallest labels are $A + G$ and B.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named B (highest label of the two) and the right child is the tree $A + G$ (lowest label of the two).

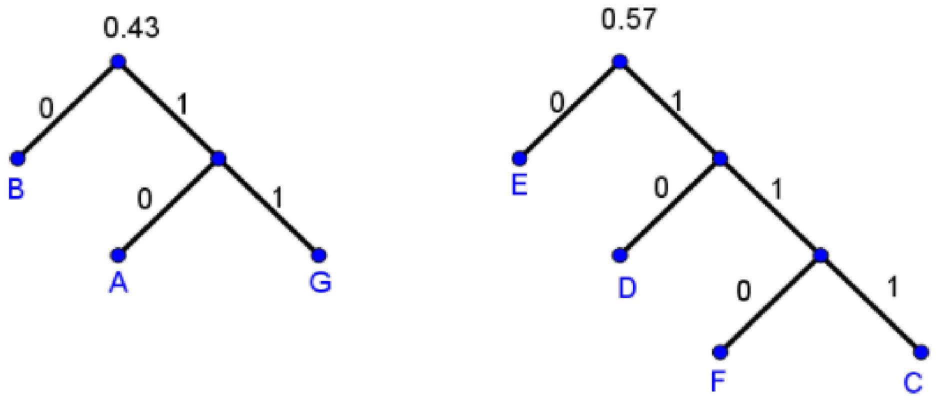
The label of the new tree is the sum of the labels of the trees that were replaced.



We then note that the trees with the smallest labels are E and $D + F + C$.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named E (highest label of the two) and the right child is the tree $D + F + C$ (lowest label of the two).

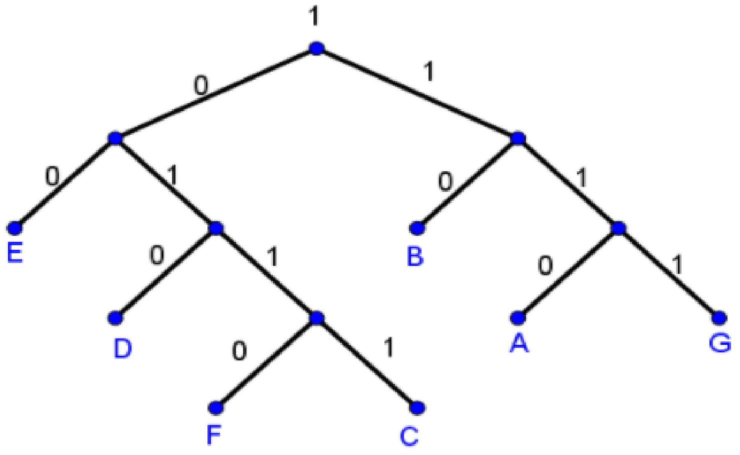
The label of the new tree is the sum of the labels of the trees that were replaced.



We then note that the trees with the smallest labels are $B + A + G$ and $E + D + F + C$.

We will then replace their trees with a new tree that has a root with a left and right child. The edge to the left child is labeled 0 and the edge to the right child is labeled 1. The left child is named $E + D + F + C$ (highest label of the two) and the right child is the tree $B + A + G$ (lowest label of the two).

The label of the new tree is the sum of the labels of the trees that were replaced.



The encoding of the letters is then the sequence of labels of the edges in the path from the root to the letter.

$$\begin{aligned} A &: 110 \\ B &: 10 \\ C &: 0111 \\ D &: 010 \\ E &: 00 \\ F &: 0110 \\ G &: 111 \end{aligned}$$

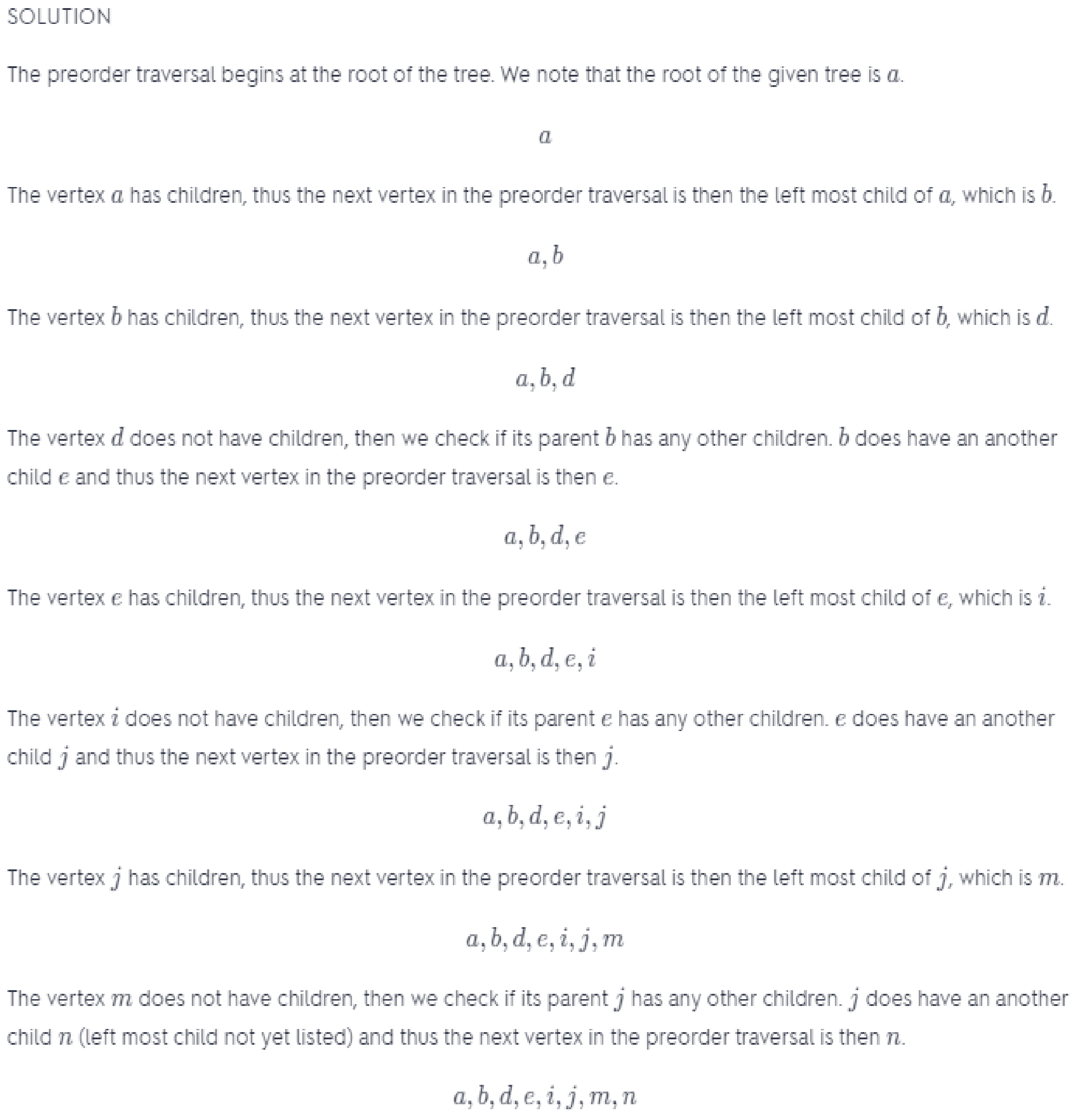
The weight of the letter is the number of bits in the code of the letter

$$\begin{aligned} w_A &: 3 \\ w_B &: 2 \\ w_C &: 4 \\ w_D &: 3 \\ w_E &: 2 \\ w_F &: 4 \\ w_G &: 3 \end{aligned}$$

The average number of bits required is then the sum of the products of the weights and the frequencies.

$$\begin{aligned} \text{Average number of bits required} &= \sum_{i \in \{A,B,C,D,E,F,G\}} w_i \cdot f_i \\ &= 3 \cdot 0.10 + 2 \cdot 0.25 + 4 \cdot 0.05 + 3 \cdot 0.15 + 2 \cdot 0.30 + 4 \cdot 0.07 + 3 \cdot 0.08 \\ &= 2.57 \end{aligned}$$

11.3(8)



SOLUTION

The preorder traversal begins at the root of the tree. We note that the root of the given tree is a.

$$a$$

The vertex a has children, thus the next vertex in the preorder traversal is then the left most child of a, which is b.

$$a, b$$

The vertex b has children, thus the next vertex in the preorder traversal is then the left most child of b, which is d.

$$a, b, d$$

The vertex d does not have children, then we check if its parent b has any other children. b does have an another child e and thus the next vertex in the preorder traversal is then e.

$$a, b, d, e$$

The vertex e has children, thus the next vertex in the preorder traversal is then the left most child of e, which is i.

$$a, b, d, e, i$$

The vertex i does not have children, then we check if its parent e has any other children. e does have an another child j and thus the next vertex in the preorder traversal is then j.

$$a, b, d, e, i, j$$

The vertex j has children, thus the next vertex in the preorder traversal is then the left most child of j, which is m.

$$a, b, d, e, i, j, m$$

The vertex m does not have children, then we check if its parent j has any other children. j does have an another child n (left most child not yet listed) and thus the next vertex in the preorder traversal is then n.

$$a, b, d, e, i, j, m, n$$

The vertex n does not have children, then we check if its parent j has any other children. j does have an another child o and thus the next vertex in the preorder traversal is then o.

$$a, b, d, e, i, j, m, n, o$$

The vertex o does not have children, then we check if its parent j has any other children not yet listed in the preorder traversal.

All children of j are already mentioned in the preorder traversal, then we check if its parent e has any other children not yet listed in the preorder traversal.

All children of e are already mentioned in the preorder traversal, then we check if its parent b has any other children not yet listed in the preorder traversal.

All children of b are already mentioned in the preorder traversal, then we check if its parent a has any other children not yet listed in the preorder traversal.

a does have an another child c and thus the next vertex in the preorder traversal is then c.

$$a, b, d, e, i, j, m, n, o, c$$

The vertex c has children, thus the next vertex in the preorder traversal is then the left most child of f, which is f.

$$a, b, d, e, i, j, m, n, o, c, f$$

The vertex f does not have children, then we check if its parent c has any other children. c does have an another child g (left most child not yet listed) and thus the next vertex in the preorder traversal is then g.

$$a, b, d, e, i, j, m, n, o, c, f, g$$

The vertex g does not have children, then we check if its parent c has any other children. c does have an another child h (left most child not yet listed) and thus the next vertex in the preorder traversal is then h.

$$a, b, d, e, i, j, m, n, o, c, f, g, h$$

The vertex h has children, thus the next vertex in the preorder traversal is then the left most child of k, which is f.

$$a, b, d, e, i, j, m, n, o, c, f, g, h, k$$

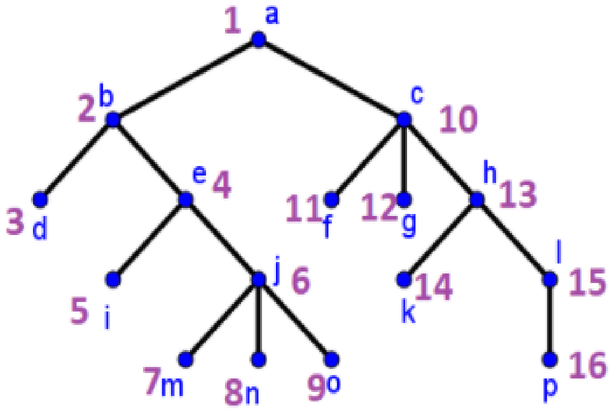
The vertex k does not have children, then we check if its parent h has any other children. h does have an another child l and thus the next vertex in the preorder traversal is then l.

$$a, b, d, e, i, j, m, n, o, c, f, g, h, k, l$$

The vertex l has children, thus the next vertex in the preorder traversal is then the left most child of l, which is p.

$$a, b, d, e, i, j, m, n, o, c, f, g, h, k, l, p$$

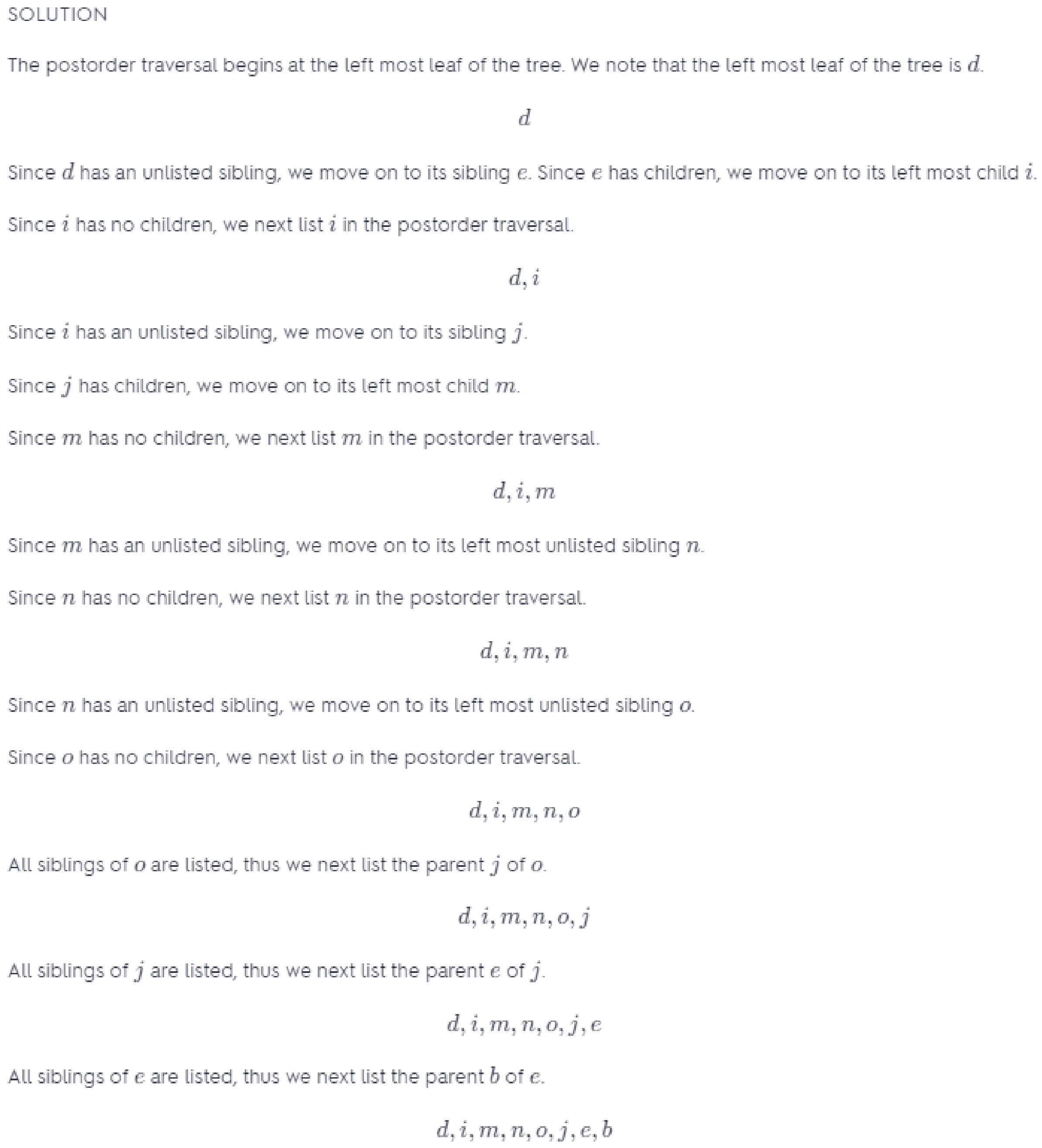
We now note that all vertices of the tree are mentioned in the preorder traversal, thus the preorder traversal of the tree is $a, b, d, e, i, j, m, n, o, c, f, g, h, k, l, p$.



Result

$$a, b, d, e, i, j, m, n, o, c, f, g, h, k, l, p$$

11.3(14)



SOLUTION

The postorder traversal begins at the left most leaf of the tree. We note that the left most leaf of the tree is d.

$$d$$

Since d has an unlisted sibling, we move on to its sibling e. Since e has children, we move on to its left most child i.

Since i has no children, we next list i in the postorder traversal.

$$d, i$$

Since i has an unlisted sibling, we move on to its sibling j.

Since j has children, we move on to its left most child m.

Since m has no children, we next list m in the postorder traversal.

$$d, i, m$$

Since m has an unlisted sibling, we move on to its left most unlisted sibling n.

Since n has no children, we next list n in the postorder traversal.

$$d, i, m, n$$

Since n has an unlisted sibling, we move on to its left most unlisted sibling o.

Since o has no children, we next list o in the postorder traversal.

$$d, i, m, n, o$$

All siblings of o are listed, thus we next list the parent j of o.

$$d, i, m, n, o, j$$

All siblings of j are listed, thus we next list the parent e of j.

$$d, i, m, n, o, j, e$$

All siblings of e are listed, thus we next list the parent b of e.

$$d, i, m, n, o, j, e, b$$

Since b has an unlisted sibling, we move on to its left most unlisted sibling c.

Since c has children, we move on to its left most child f.

Since f has no children, we next list f in the postorder traversal.

$$d, i, m, n, o, j, e, b, f$$

Since f has an unlisted sibling, we move on to its left most unlisted sibling g.

Since g has no children, we next list g in the postorder traversal.

$$d, i, m, n, o, j, e, b, f, g$$

Since g has an unlisted sibling, we move on to its left most unlisted sibling h.

Since h has children, we move on to its left most child k.

Since k has no children, we next list k in the postorder traversal.

$$d, i, m, n, o, j, e, b, f, g, k$$

Since k has an unlisted sibling, we move on to its left most unlisted sibling l.

Since l has children, we move on to its left most child p.

Since p has no children, we next list p in the postorder traversal.

$$d, i, m, n, o, j, e, b, f, g, k, p$$

p does not have siblings, thus we next list the parent l of p.

$$d, i, m, n, o, j, e, b, f, g, k, p, l$$

All siblings of l are listed, thus we next list the parent h of l.

$$d, i, m, n, o, j, e, b, f, g, k, p, l, h$$

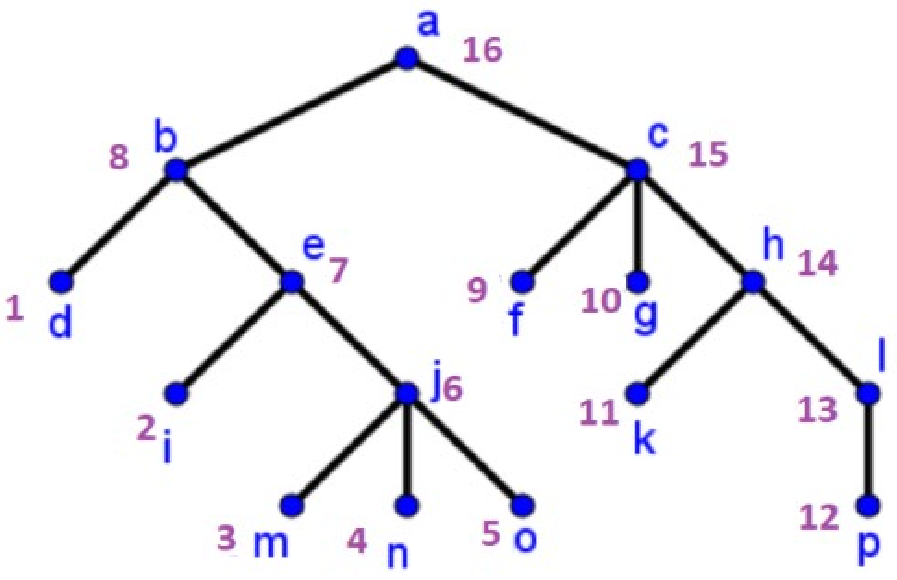
All siblings of h are listed, thus we next list the parent c of h.

$$d, i, m, n, o, j, e, b, f, g, k, p, l, h, c$$

All siblings of c are listed, thus we next list the parent a of c.

$$d, i, m, n, o, j, e, b, f, g, k, p, l, h, c, a$$

We now note that all vertices of the tree are mentioned in the postorder traversal, thus the postorder traversal of the tree is $d, i, m, n, o, j, e, b, f, g, k, p, l, h, c, a$.



Result

$$d, i, m, n, o, j, e, b, f, g, k, p, l, h, c, a$$

11.3(23.b)

(b)

$$\uparrow - * 3\ 3 * 4\ 2\ 5$$

Start from the right most operation symbol.

For every operation symbol, the two integers following the operation symbol are the symbols on which the operation needs to be performed. Then replace the operation symbol with the two following integers by the result of the operation

Then we repeat for the right most operation symbol in the remaining string.

$$\uparrow\ -\ *\ 3\ 3\ \underbrace{*\ 4\ 2}_{4*2=8}\ 5$$

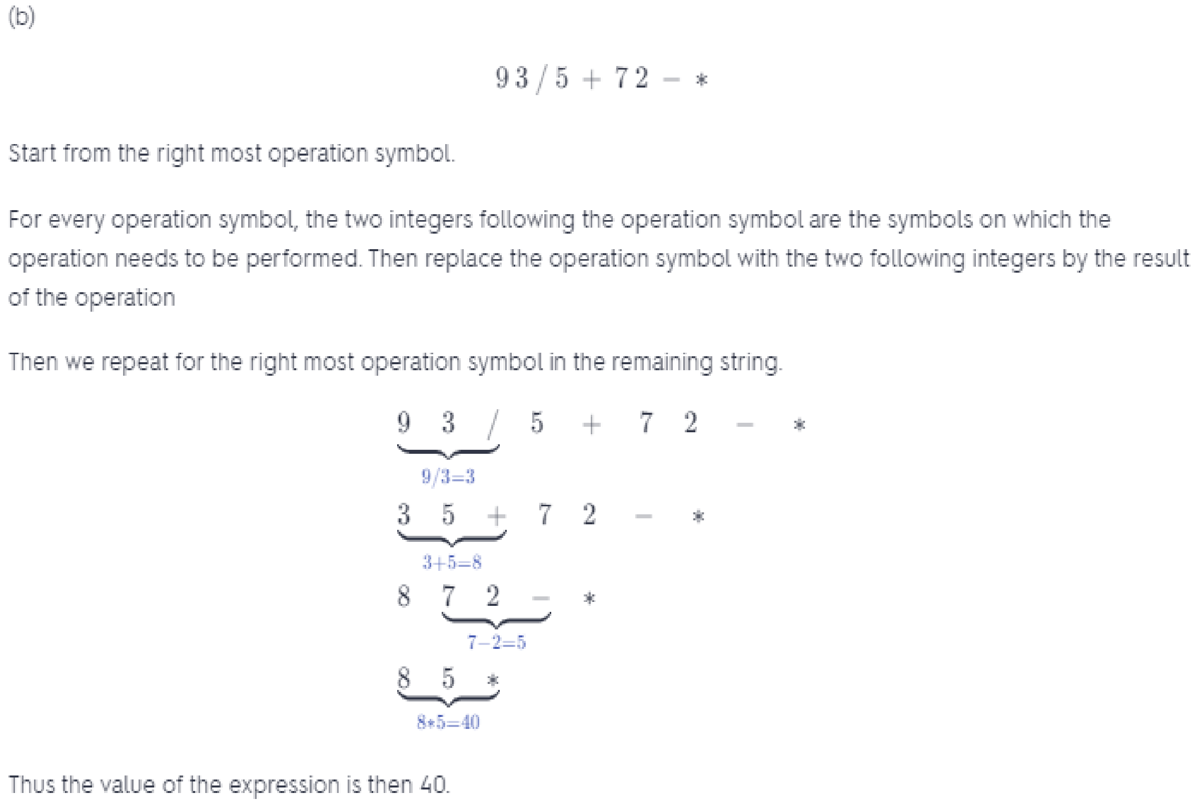
$$\uparrow\ -\ \underbrace{*\ 3\ 3}_{3*3=9}\ 8\ 5$$

$$\uparrow\ \underbrace{-\ 9\ 8}_{9-8=1}\ 5$$

$$\underbrace{\uparrow\ 1\ 5}_{1\uparrow 5=1^5=1}$$

Thus the value of the expression is then 1.

11.3(24.b)



(b)

$$9\,3\,/\,5\,+\,7\,2\,-\,*$$

Start from the right most operation symbol.

For every operation symbol, the two integers following the operation symbol are the symbols on which the operation needs to be performed. Then replace the operation symbol with the two following integers by the result of the operation

Then we repeat for the right most operation symbol in the remaining string.

$$\underbrace{9\quad 3\quad /}_{9/3=3}\quad 5\quad +\quad 7\quad 2\quad -\quad *$$
$$\underbrace{3\quad 5\quad +}_{3+5=8}\quad 7\quad 2\quad -\quad *$$
$$8\quad \underbrace{7\quad 2\quad -}_{7-2=5}\quad *$$
$$\underbrace{8\quad 5\quad *}_{8*5=40}$$

Thus the value of the expression is then 40.