Assignment 4 - CS 4071 - Spring 2018

Due: 2018-02-19

Group #13: Robert DiMartino (dimartrt), Hayden Schiff (schiffha), Jeremiah Leak (leakjz)

1. Exercise 6.5

Problem: Solve the following instance of the knapsack problem for capacity C=30.

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
v_i	60	50	40	30	20	10	5	1
w_i	30	100	10	10	8	8	1	1

To solve this knapsack problem, we want to greedily select objects of decreasing densities, i.e. the ratio of value to weight, v_i/w_i, until the knapsack is full. So first we compute the densities for each object.

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$d_i = v_i/w_i$	2	0.5	4	3	2.5	1.25	5	1

Then we repeatedly select available objects with the next highest density until the knapsack is full (or we run out of items).

0. Starting capacity: 30
1. Select: 6 ($d = 5, w = 1$), fraction: 1, remaining capacity: 29
2. Select: 2 ($d = 4, w = 10$), fraction: 1, remaining capacity: 19
3. Select: 3 ($d = 3, w = 10$), fraction: 1, remaining capacity: 9
4. Select: 4 ($d = 2.5, w = 8$), fraction: 1, remaining capacity: 1
5. Select: 0 ($d = 2, w = 30$), fraction: 1/30, remaining capacity: 0

The fractions, f_i, of each item that yield the optimal value for the knapsack are provided in the table below.

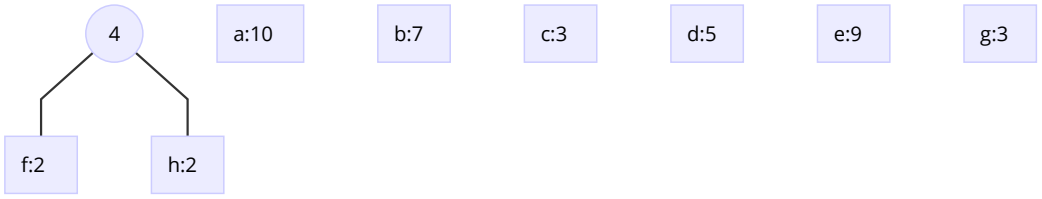
i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Total**
f_i	1/30	0	1	1	1	0	1	0	
$f_i v_i$	2	0	40	30	20	0	5	0	97
$f_i w_i$	1	0	10	10	8	0	1	0	30

2. Exercise 6.9

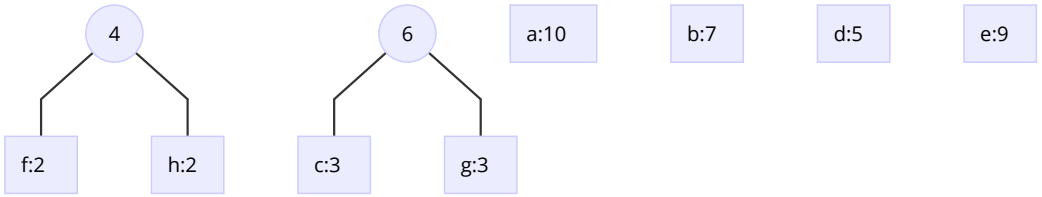
Problem: Trace the action of *HuffmanCode* for the letters *a, b, c, d, e, f, g, h* occuring with frequencies 10, 7, 3, 5, 9, 2, 3, 2.

0. Initial forest

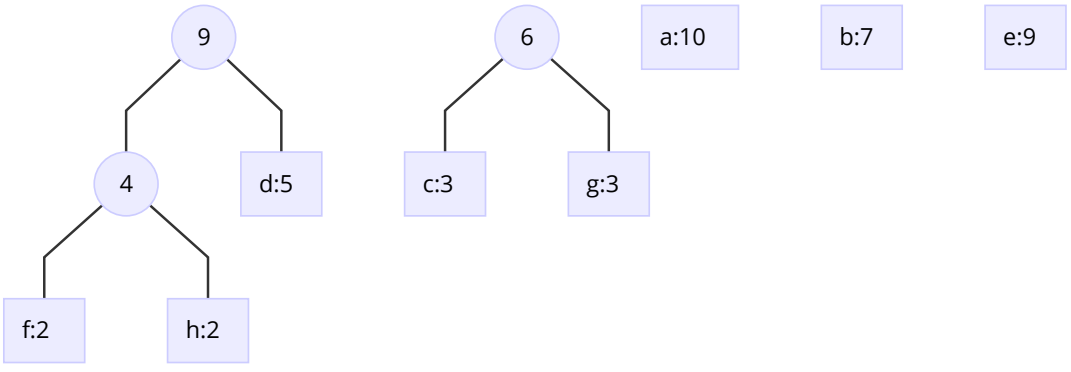
1. Stage 1



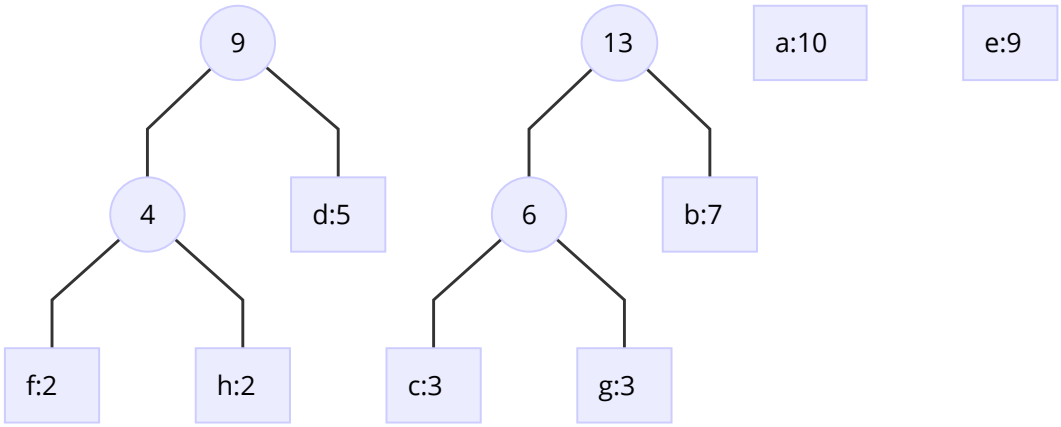
2. Stage 2



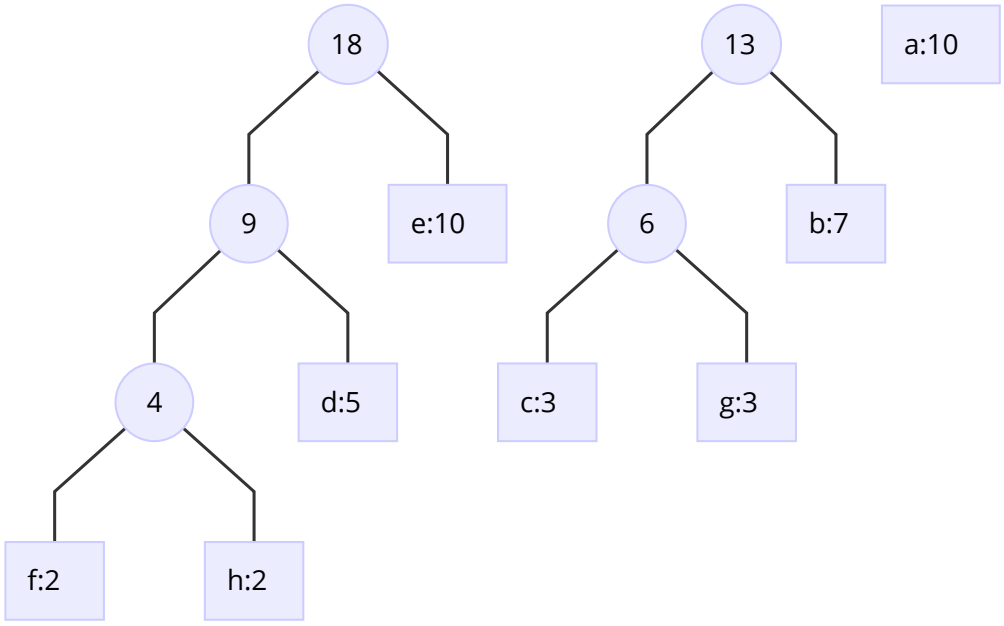
3. Stage 3



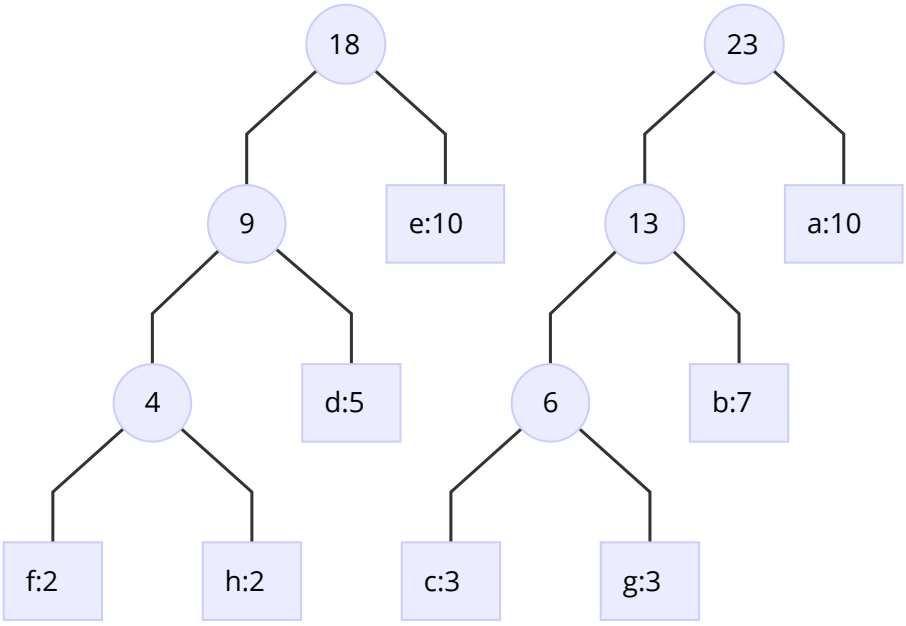
4. Stage 4



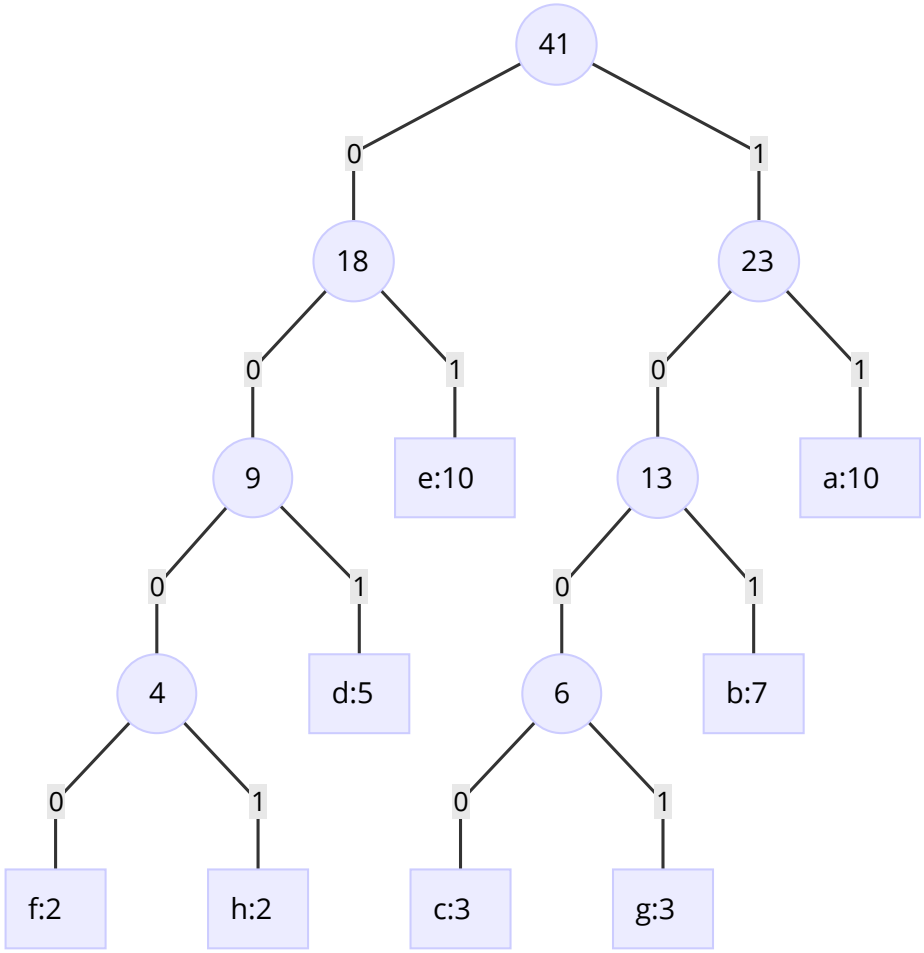
5. Stage 5



6. Stage 6



7. Stage 7: Huffman Tree



letter	frequency	encoding
a	10	`11`
b	7	`101`
c	3	`1000`
d	5	`001`
e	9	`01`
f	2	`0000`
g	3	`1001`
h	2	`0001`

3. Exercise 6.10

Problem: Given the Huffman Code Tree in Figure 6.6, decode the string `100111100001101111001`

> **Figure 6.6**
>
> | letter | frequency | encoding |
> |---|---|---|
> | *a* | 9 | `01` |
> | *b* | 8 | `00` |
> | *c* | 5 | `101` |
> | *d* | 3 | `1001` |
> | *e* | 15 | `11` |
> | *f* | 2 | `1000` |

We read binary characters from the encoded string one at a time. Because the Huffman Code is a prefix code, when the binary substring we've read matches one of the letter encodings from Figure 6.6 we can unambiguously determine which letter is decoded.

Reading the string from the front:

1. `1001` = *d*
2. `11` = *e*
3. `1000` = *f*
4. `01` = *a*
5. `101` = *c*
6. `11` = *e*
7. `1001` = *d*

Then `100111100001101111001` decoded is "defaced".