Necole Goodman

CSE 100 Midterm 3

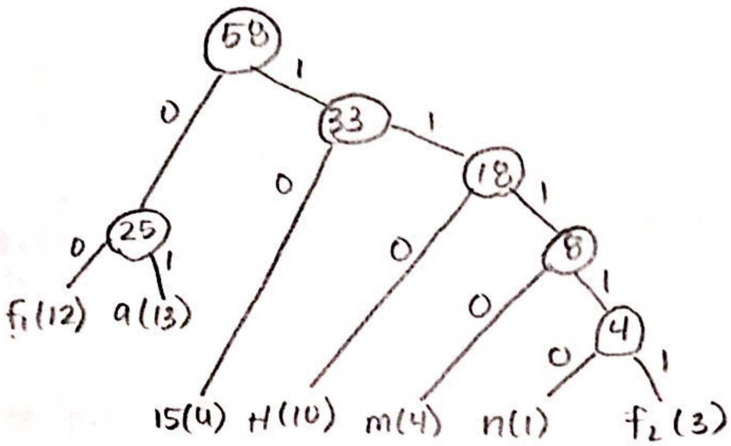
1. This statement is True. A code is considered a prefix-free code if no codeword is a prefix of another one. In this particular case, every leaf must have a sibling. An optimal code for a file is always represented by a full binary tree, where every non leaf node has two children. If there aren't any siblings where the codewords only start w/ 10 and none start w/ 11, for example then we don't have a full tree.

Scanned with CamScanner

2.

H	u	f_1	f_2	m	a	n
10	15	12	3	4	13	1

n	f_2	m	H	f_1	a	u
1	3	4	10	12	13	15



char	code	freq	length
H	110	10	30
u	10	15	30
f_1	00	12	24
f_2	11111	3	15
m	1110	4	16
a	01	13	26
n	11110	1	5

∴ Both f_2 and n have at least code in length 5.

3. $r_j = \begin{cases} \max_{1 \le i \le j} (p_i + r_{j-i}) & \text{if } j \ge 1 \\ 0 & \text{otherwise } (j = 0) \end{cases}$

more efficient.

$r_j = \begin{cases} \max\{p_j, \max_{1 \le i \le \lfloor j/2 \rfloor} (r_i + r_{j-i})\} & \text{if } j \ge 2 \\ p_1 & \text{otherwise } (j = 1) \end{cases}$

To prove if the recursion is correct I will plug in values of 'j'.

if $j = 1$ (given) the price of rod = the max revenue

then $r_j = P_1$

$r_1 = P_1$

$i = \frac{j}{2} = \frac{2}{2} = 1$

if $j = 2$

then $r_2 = \max \{ P_2, \max(r_1 + r_{2-1})$

$r_2 = \max \{ P_2, \max(r_1 + r_1) \leftarrow$ since $r_1 = P_1$

$r_2 = \max \{ P_2, \max(P_1 + P_1)$

$r_2 = \max \{ P_2, \max(P_2)$

Yes the recursion works.

4. $M(i) = \begin{cases} 0 & \text{if } i = 0 \quad \text{(base case)} \\ \max\{M(j) + w_i, M(i-1)\} & \text{otherwise} \end{cases}$

```
if i==0          // The first condition of the iteration
  return 0

if m[i] = m[i]
  return m[i]

int x = (i-1 ,j)         // initializing variables
int y = ( p[i], j]

if (x>y)           // making the comparison
  m[j] = x
else m[j] = y
return m[j]
```

5. m[i,j] = max # of mults needed to compute the product
s[i,j] = k implies there is an optimal solution.

must use k to find m[1][n]

```
for x = 0 to i
  for y = 0 to j

    if s[i,j] = 0
       m[i,j] = 0

    if s[i,j] = k
    m[i,j] = m[i,k] + m[k+1,j] + (# of columns in matrix)
end
  end
return m[1,n]
```

runtime: $O(n^2)$

6. For this implementation, I will use the recursive Top-down implementation method pseudocode w/ minor modifications

```
Cut-Rod (P,n)
  if n == 0
     return 0
  q = -∞ (lowest int)
                                          values given (k)
                                               ↓           ↙ k^2 dollars
  for i = 1 to n
     q = max (q, P[i] + cut-rod (p,n-i) - P[i]^2)
  return q
```

The run time of cut-rod is exponential in n. since cut-rod considers all the 2^{n-1} ways of cutting up a rod of length n.