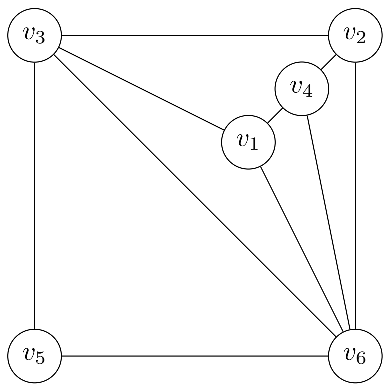
VE203 Assignment 10

Liu Yihao 515370910207

Exercise 10.1

i) Since there isn't a circuit with three vertices, and $e = 14$, $v = 8$, $e > 2v - 4$, so it is not a planner graph.

ii)



iii) Since $e = 18$, $v = 7$, $e > 3v - 6$, so it is not a planner graph.

Exercise 10.2

Since it is a full m-*ary* balanced tree of height h, it must contain a perfect m-*ary* tree of height $h - 1$, which has m^{h-1} leaves. If there are $km, k \in [1, m] \cap N$ leaves of height h, then there are $m^{h-1} + k(m-1)$ leaves, which is more than m^{h-1} leaves.
Since $l \in [m^{h-1} + (m-1), m^h]$, and a full m-*ary* balanced tree of height $h - 1$ has at most m^{h-1} leaves, a full m-*ary* balanced tree of height $h + 1$ has at least $m^h + (m - 1)$ leaves, $h = [\log_m l]$.

Exercise 10.3

At most three weighings are needed.

Input: Four coins with weight m_1, m_2, m_3, m_4 (unknown), one of them may be counterfeit
Output: The counterfeit coin and whether it is heavier or lighter (if exists)

Compare m_1 and m_2
Compare m_3 and m_4
if $m_1 \neq m_2$ **then**
    Compare $min(m_1, m_2)$ and m_3
    if $min(m_1, m_2) = m_3$ **then**
        $max(m_1, m_2)$ is counterfeit (heavier)
    else
        $min(m_1, m_2)$ is counterfeit (lighter)
    end if
else if $m_3 \neq m_4$ **then**
    Compare $min(m_3, m_4)$ and m_1
    if $min(m_3, m_4) = m_1$ **then**
        $max(m_3, m_4)$ is counterfeit (heavier)
    else
        $min(m_3, m_4)$ is counterfeit (lighter)
    end if
else
    No coin is counterfeit
end if

Exercise 10.4

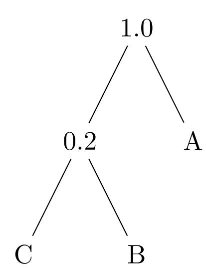
If there is n symbols $a_1, ..., a_n$ with frequencies $p_1, ..., p_n$, whose bits are $b_1, ..., b_n$. If $p_1 > ... > p_n$ and $b_1 < ... < b_n$, according to sequence inequality, we can obtain that

$$\sum_{i=1}^{n} p_i b_i \leqslant \sum_{i=1, j=f(i)}^{n} p_i b_j$$

where $i, j \in [1, n] \cap N$ and $f(i)$ is a random bijective function from i to j Since the Huffman codes use the fewest bits for the biggest frequency, which satisfy the model above, it uses the fewest bits in total.

Exercise 10.5

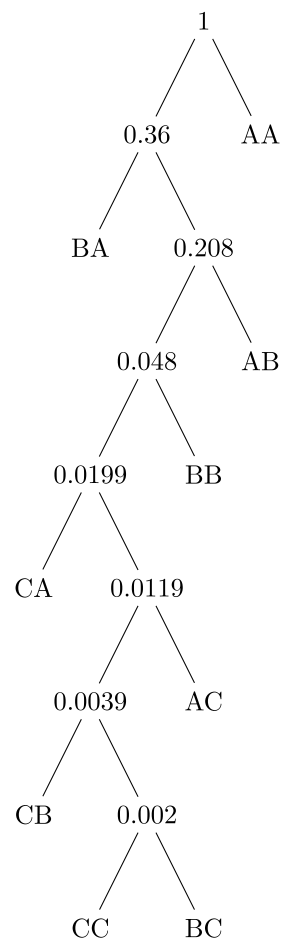
i)



$$C : 00, \quad B : 01, \quad A : 1$$

ii)

$$AA : 0.64, \quad AB : 0.152, \quad AC : 0.008$$
$$BA : 0.152, \quad BB : 0.0361, \quad BC : 0.0019$$
$$CA : 0.008, \quad CB : 0.0019, \quad CC : 0.0001$$



$$AA : 1, \quad AB : 011, \quad AC : 010011$$
$$BA : 00, \quad BB : 0101, \quad BC : 01001011$$
$$CA : 01000, \quad CB : 0100100, \quad CC : 01001010$$

iii) In part i),

$$\overline{N} = 1 \times 0.8 + 2 \times 0.19 + 2 \times 0.01 = 1.2$$

In part ii),

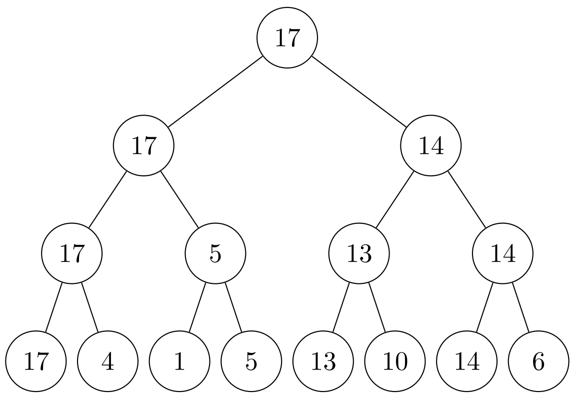
$$\overline{N} = \frac{1}{2}\Big[1\times0.64+(2+3)\times0.152+4\times0.0361+(5+6)\times0.008+(7+8\times0.0019)+8\times0.0001\Big] = 0.8489$$

So part ii) is more efficient.

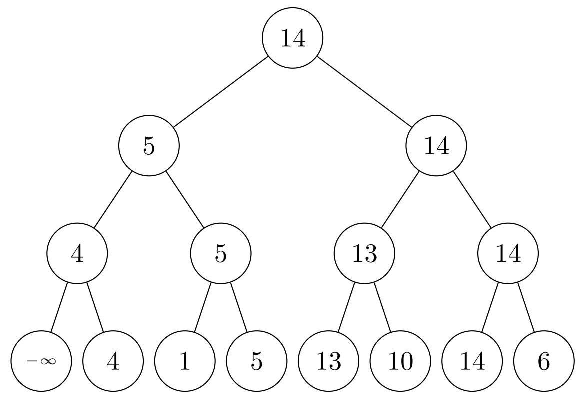
Exercise 10.6

i)

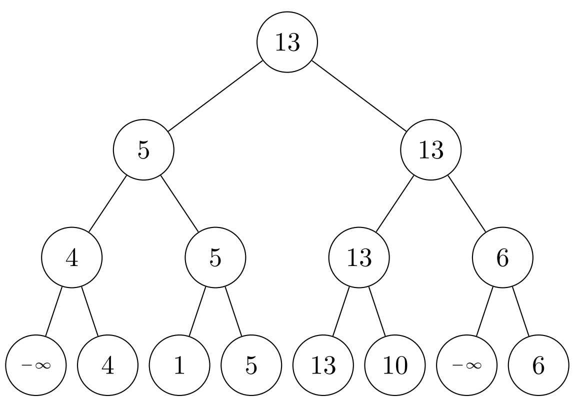
Step 1:



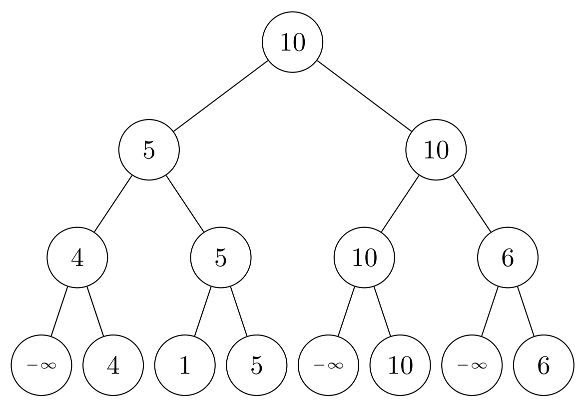
Step 2:



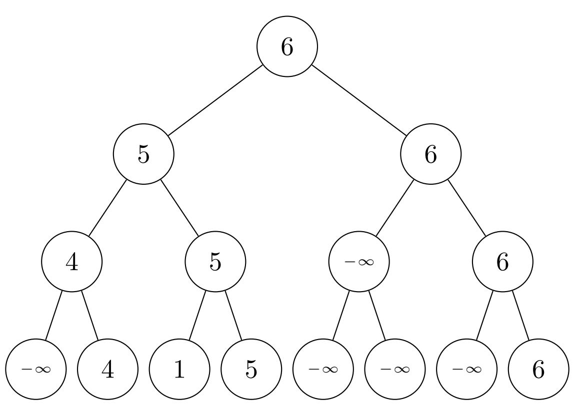
Step 3:



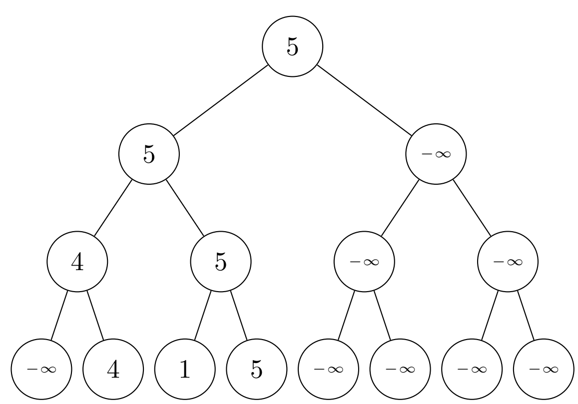
Step 4:



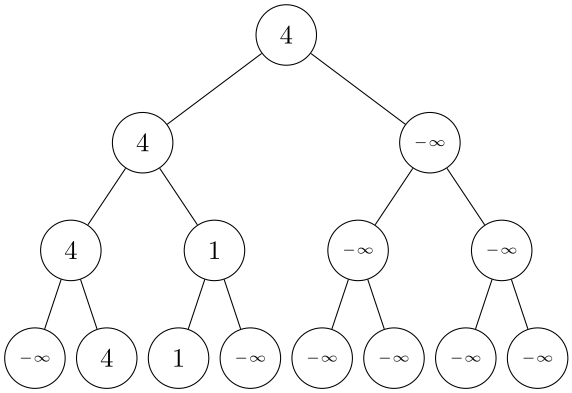
Step 5:



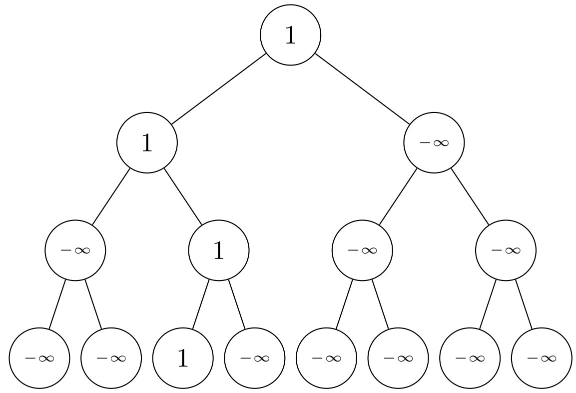
Step 6:



Step 7:



Step 8:



ii)

$$C = \sum_{i=0}^{k-1} 2^i = 2^k - 1 = n - 1$$

iii) When searching downwards, there are $\log_2 n$ comparisons to find the larger node.
When backtracking, there are $\log_2 n$ comparisons to update the node.
So there are $2\log_2 n$ comparisons in total.

iv)

$$O(n - 1 + 2(n-1)\log_2 n) = O(n \log n)$$