Homework 9

Lei Zhang

April 27, 2016

1 Exercises 9.4 - 1

a.

symbol	A	B	C	D	-
frequency	0.4	0.1	0.2	0.15	0.15
codeword	0	100	111	101	110

b.

According to the codewords above, ABACABAD will be encoded as 0100011101000101.

c.

According to the codewords above, 100010111001010 will be decoded as BAD_ADA.

2 Exercises 9.4 - 9

3 Exercises 10.1 - 2

a.

b.

4 Exercises 10.1 - 10

a.

primal problem:
maximize cx
subject to $Ax \leq b$, $X \geq 0$
where :

$$c = [c_1 ... c_n], x = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix}, A = \begin{bmatrix} a_{11} & ... & a_{1n} \\ . & & . \\ . & & . \\ . & & . \\ a_{m1} & ... & a_{mn} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ . \\ . \\ . \\ b_m \end{bmatrix}$$

dual problem:
maximize $b^T x$
subject to $A^T y \geq c^T$, $y \geq 0$
where b, A, c, y, x are defined above in **a**.

b.

minimize $6y_1 + 2y_2$
subject to :

$$y_1 + y_2 \geq 1$$
$$y_1 - y_2 \geq 4$$
$$y_1 - 2y_2 \geq -1$$
$$y_1, y_2 \geq 0$$

c.

primal problem:
Standard form:
maximize $x_1 + 4x_2 - x_3$
subject to
$x_1 + x_2 + x_3 + u = 6$
$x_1 - x_2 - 2x_3 + v = 2$
$x_1, x_2, x_3, u, v \geq 0$
The simplex tableau:

	x_1	x_2	x_3	u	v	
u	1	1	1	1	0	6
v	1	-1	-2	0	1	2
	-1	-4	1	0	0	0

	x_1	x_2	x_3	u	v	
u	1	1	1	1	0	6
v	2	0	-1	1	1	8
	3	0	5	4	0	24

The optimal solution is $x_1 = 0, x_2 = 6, x_3 = 0$
dual problem:

The optimal solution is $y_1 = 4, y_2 = 0$
The results of objective functions in the primal aand dual problems are equal.

5 Exercises 10.2 - 2

6 Exercises 10.3 - 1

a.

b.

No augmentation exists.

In part b, vertices 5 and 8 and linked and only linked to vertice 4, which means only one of them can be matched. As there are already three links, there won't be one more augmentation.

7 Exercises 10.3 - 4

a.

For $V = \{1, 2, 3, 4\}$, when $S = \{1, 2, 3, 4\}$, $|R(S)| = 3$ and $|S| = 4$, so there's no matching that matches all vertices of the set V.

For $V = \{5, 6, 7\}$, every $|R(S)|$ is greater than $|S|$, so there's a mathing that matches all vertices of the set V.

b.

```
//Input: A bipartite graph G =< V, U, E >
//Output: returns yes if there is a matching in G that matches all vertices in
  V and returns no otherwise.
  function IsMaximumMatching(G)
      while notEmpty(Q) do
          w ← Front(Q); Dequeue(Q)
          if w ∈ V then
              for every vertex u adjacent to w do
                  if u is free then
                      M ← M ∪ (w, u)
                      v ← w
                      while v is labeled do
                          u ← vertex indicated by v's label; M ← M − (v, u)
                          v ← vertex indicated by u's label; M ← M ∪ (v, u)
                      end while
                      remove all vertex labels
                      reinitialize Q with all free vertices in V
                      Break
                  else
                      if w, u ∉ M and u is unlabeled then
```

```
                label u with w
                Enqueue(Q, u)
            end if
        end if
      end for
    else
      label the mate v of w with w
      Enqueue(Q, v)
    end if
  end while
  if |M| = |V| then
    return yes
  else
    return no
  end if
end function
```

The time efficiency of the algorithm based on Hall's Marriage Theorem is too expensive, so I'd like to to use the algorithm shown above.