name: kevin wong
filename: Math115 homework12
date: 04/20/2022
desc: https://courses.csail.mit.edu/6.042/spring18/mcs.pdf (Links to an external site.) please do these problems: additional problem, 12.23, 12.56ab, 12.57, 12.63.

Character Frequency Problem

M115 Homework12, Spring 2022.

1. Given the character frequencies

$$(a, b, c, d, e, f, g) = (0.26, 0.02, 0.09, 0.08, 0.30, 0.14, 0.11)$$

I tried encoding the characters in a binary tree shown on the right.

a) With this encoding, what is the bit-string translation of the word 'bead' ?

b) With this encoding, what is the average number of bits used per character?

c) If possible, find an encoding that has a lower average number than your answer to part (b).

- a) $b = 0100, e = 11, a = 10, d = 0110$

  bead is encoded with bit string 0100 1110 0110.

- b) The average number of bits used per character is $2(0.26) + 4(0.02) + 4(0.09) + 4(0.08) + 2(0.30) + 2(0.14) + 4(0.11) = 2.6$

- c) Using Hoffman coding to get the optimal encoding for characters into bit strings given their frequencies in a text.

  Heavier weight goes on left.

  Lighter weight on right.

  Branching left is assigned a zero bit.

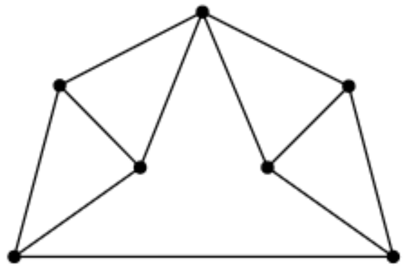
  Branching right is assigned a one bit.

```
a: 0.26    b: 0.02    c : 0.09    d : 0.08    e : 0.30    f : 0.14    g : 0.11

a: 0.26    cb: 0.11    d : 0.08    e : 0.30    f : 0.14    g : 0.11
          /\
         c  b

a: 0.26    cbd: 0.19    e : 0.30    f : 0.14    g : 0.11
           /\
          /\ d
         c  b

a: 0.26    cbd: 0.19    e : 0.30    fg : 0.25
           /\                        /\
          /\ d                      f  g
         c  b

a: 0.26    fgcbd: 0.44    e : 0.30
           / \
          /   \
         /     \
        /\     /\
       f  g   /\  d
             c  b

          fgcbd: 0.44    ea : 0.56
           / \            /\
          /   \          e  a
         /     \
        /\     /\
       f  g   /\  d
             c  b

          eafgcbd: 1
            /  \
          0/    \1
          /      \
        0/\1    / \
        e  a  0/   \1
              /     \
            0/\1   0/ \1
            f  g  0/\1 \
                  c  b  d
```

A lower average number of bits used per character is $2(0.26)+4(0.02)+4(0.09)+3(0.08)+2(0.30)+3(0.14)+3(0.11)=2.55$

12.23

Problem 12.23.
Let G be the graph below.[16] Carefully explain why $\chi(G) = 4$.



I think it's logical to work from a smaller chromatic number, $\chi(G)$ to a larger one. This graph has more than one edge, we know $\chi(G) \neq 1$. We can also conclude that $\chi(G) \neq 2$ because in this graph there exist a triangle and a pentagon, and we know $\chi(Cycle_{odd}) = 3$. If the most bottom edge wasn't there, the chromatic number of 3 ($\chi(G) = 3$) would be the minimal colors needed for all adjacent vertices to not have the same colors, but since that edge exist one additional color has to be added to create a valid coloring for graph G. So $\chi(G) = 4$.

12.56ab

Problem 12.56.
Let G be the 4×4 grid with vertical and horizontal edges between neighboring vertices and edge weights as shown in Figure 12.36.

In this problem you will practice some of the ways to build minimum-weight spanning trees. For each part, list the edge weights in the order in which the edges with those weights were chosen by the given rules.

(a) Construct a minimum weight spanning tree (MST) for G by initially selecting the minimum weight edge, and then successively selecting the minimum weight edge that does not create a cycle with the previously selected edges. Stop when the selected edges form a spanning tree of G. (This is Kruskal's MST algorithm.)

For any step in Kruskal's procedure, describe a black-white coloring of the graph components so that the edge Kruskal chooses is the minimum weight "gray edge" according to Lemma 12.11.10.

(b) Grow an MST for G by starting with the tree consisting of the single vertex u and then successively adding the minimum weight edge with exactly one endpoint in the tree. Stop when the tree spans G. (This is Prim's MST algorithm.)

For any step in Prim's procedure, describe a black-white coloring of the graph components so that the edge Prim chooses is the minimum weight "gray edge" according to Lemma 12.11.10.

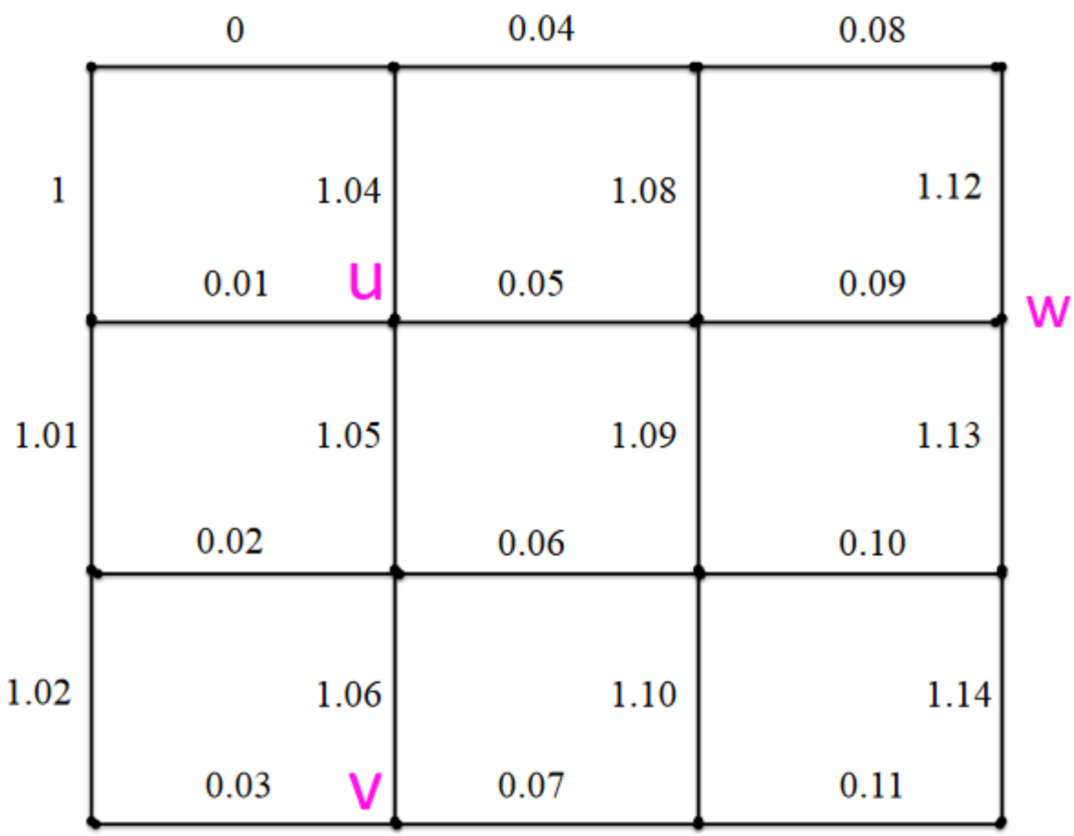


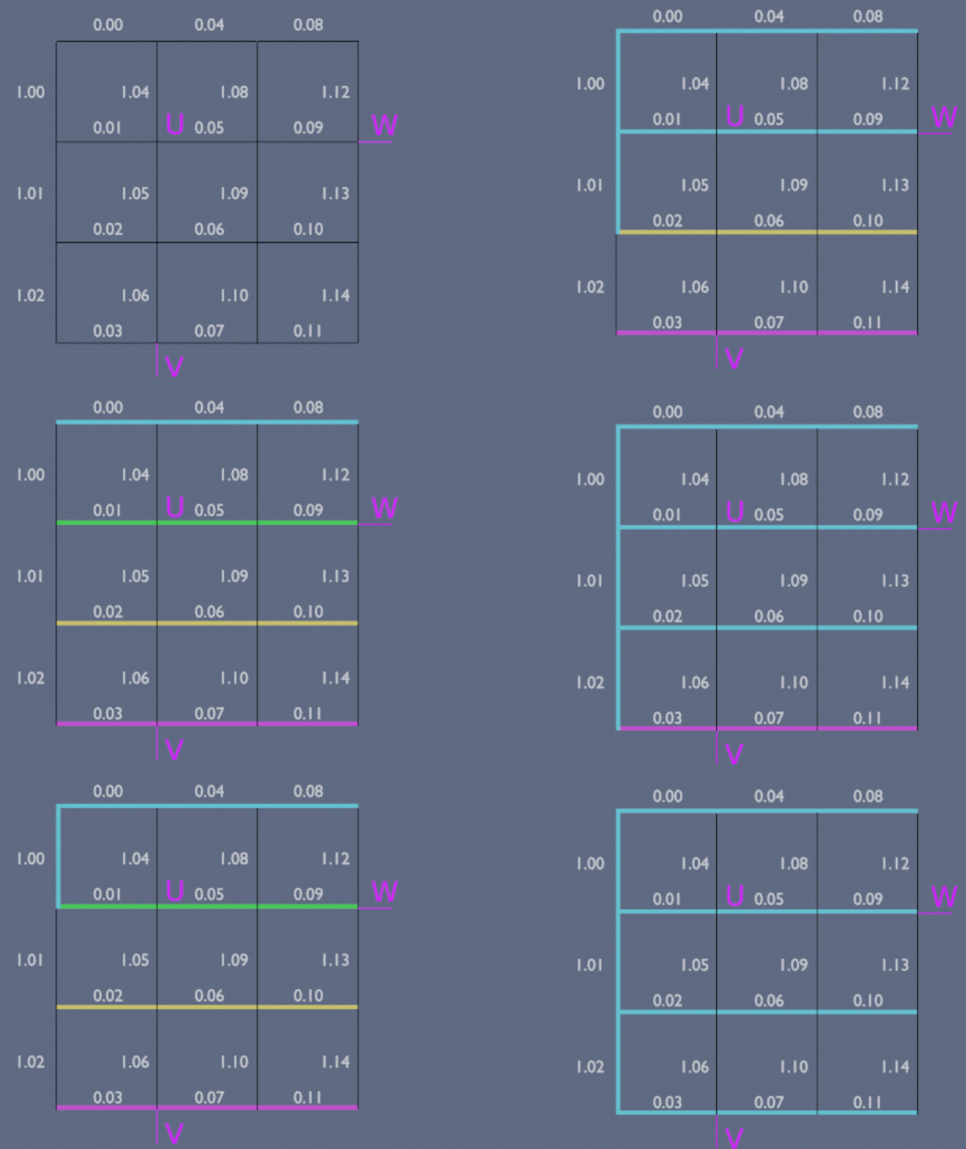
Figure 12.36 The 4x4 array graph G

- a)

Kruskals MST algorithm:

1) Start with all verts disconnected (forest).

2) Iteratively include the lightest edges while avoiding cycles, until all verts are connected.



Order the edges were selected in:
0.00, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 1.00, 1.01, 1.02.

Total weight:
0.00 + 0.01 + 0.02 + 0.03 + 0.04 + 0.05 + 0.06 + 0.07 +

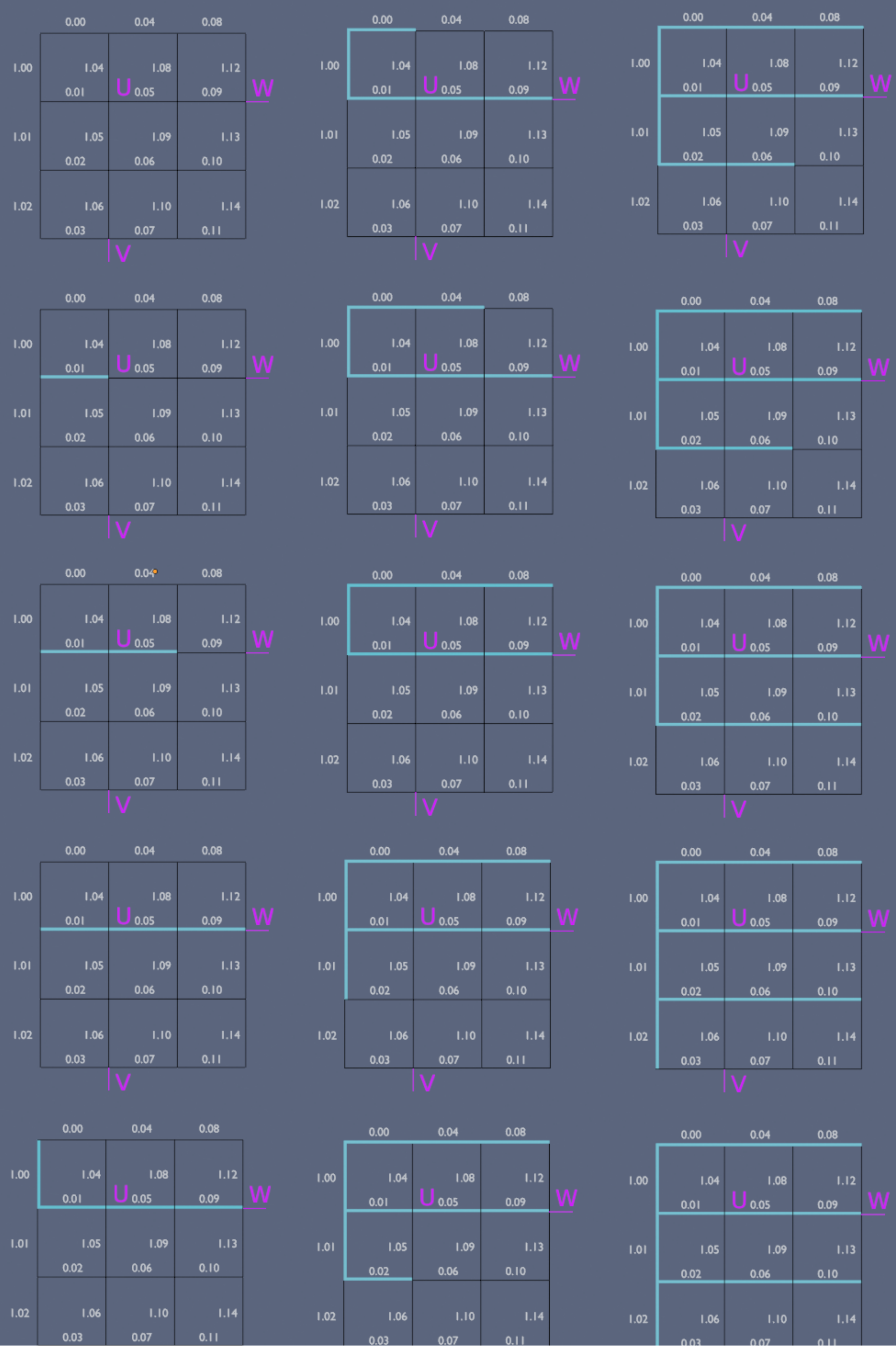
0.00 + 0.01 + 0.02 + 0.03 + 0.04 + 0.05 + 0.06 + 0.07 +
0.08 + 0.09 + 0.10 + 0.11 + 1.00 + 1.01 + 1.02 = 3.69

- b)

Prims MST algorithm

1) Start with a lightest edge(subtree with 2 verts).

2) Iteratively add the lightest possible edge that is incident to your current tree, avoiding cycles, until you have a spanning tree.



Order of edge selection:
Start at vertex U, 0.01, 0.05, 0.09, 1.00, 0.00, 0.04, 0.08, 1.01, 0.02, 0.06, 0.10, 1.02, 0.03, 0.07, 0.11.

Total Weight:
0.01 + 0.05 + 0.09 + 1.00 + 0.00 + 0.04 + 0.08 + 1.01 + 0.02 + 0.06 + 0.10 + 1.02 + 0.03 + 0.07 + 0.11 = 3.69

12.57

Problem 12.57.
In this problem you will prove:

Theorem. *A graph G is 2-colorable iff it contains no odd length closed walk.*

As usual with "iff" assertions, the proof splits into two proofs: part (a) asks you to prove that the left side of the "iff" implies the right side. The other problem parts prove that the right side implies the left.

(a) Assume the left side and prove the right side. Three to five sentences should suffice.

(b) Now assume the right side. As a first step toward proving the left side, explain why we can focus on a single connected component H within G.

(c) As a second step, explain how to 2-color any tree.

(d) Choose any 2-coloring of a spanning tree T of H. Prove that H is 2-colorable by showing that any edge *not* in T must also connect different-colored vertices.

- a) Assume the left side. Prove right side. If G is 2-colorable that means we can select any vertex in a closed cycle and assign it a color ($color_1$) and it's adjacent vertex a different color ($color_2$). As we walk along the graph we assign $color_1$ and $color_2$ in an alternating fashion until we reach a point at the end of the cycle where every vertex is assigned either $color_1$ or $color_2$ and no adjacent vertex has the same color. This is only possible if every closed cycle has an even length because an odd length closed cycle has an odd chromatic number ($\chi(C_{odd}) = 3$).✓
- b) If a connected component has an odd number closed walk then we know that cycle requires a minimum of 3 colors for a valid coloring or else two adjacent vertices would have the same color. This can be used as a counter example to disprove or prove a graph is not 2-colorable or bipartite. ✓

- c) To 2-color any tree, select two colors, color_1 and color_2. Assign the parent a color, then assign all descendants the other color. Then continuously alternate the colors each generation. This is possible because tree graphs are acyclic, meaning there doesn't exist a cycle in the graph, so we don't have to account for even or odd cycles.
- d) If all edges in T connect different colored vertices(assuming we are only using two colors), then we know T is 2-colorable and we know 2-colorable graphs have an even number of closed cycles. This is consistent with the theorum.

  If T has an edge that connects two same colored vertices or an additional third color to get a valid coloring, that means there exist an odd number closed cycle in T and that isn't consistent with what the theorum above, so the *iff* doesn't hold in these cases.✓

12.63

How many 5-vertex non-isomorphic trees are there? Explain.

I believe there are only three non-isomorphic threes.

```
Starting with the lowest degree per vertex a line:

1---2---3---4---5

    5
    |
1---2---3---4

and

    2
    |
1---5---3
    |
    4

Any kind of deletion and addition of an edge must preserve the connectedness and acyclicness
of the graph to still be a tree.
```

At this point I don't believe it is possible to find another non-isomorphic tree. Any other manipulation will lead to a connected graph, a disconnected graph, or an isomorphism. ✓

END
