Math 15 - Spring 2017 - Homework 3.5 Solutions

1. (3.5 # 14) Let L = (((15,25), (35,45)), ((50,60), (70,80))) be an SList.
   (a) Compute Search(15, L), showing all steps.

$$\begin{aligned}\texttt{Search}[15, L] &= \texttt{Search}[15, ((15,25),(35,45))] \vee \texttt{Search}[15, ((50,60),(70,80))] \\ &= \texttt{Search}[15, (15,25)] \vee \texttt{Search}[15, (35,45)] \vee \texttt{Search}[15, (50,60)] \vee \texttt{Search}[15, (70,80)] \\ &= \texttt{Search}[15, 15] \vee \texttt{Search}[15, 25] \vee \texttt{Search}[15, 35] \vee \texttt{Search}[15, 45] \vee \texttt{Search}[15, 50] \\ &\quad \vee \texttt{Search}[15, 60] \vee \texttt{Search}[15, 70] \vee \texttt{Search}[15, 80] \\ &= \texttt{true} \vee \texttt{false} \vee \texttt{false} \vee \texttt{false} \vee \texttt{false} \vee \texttt{false} \vee \texttt{false} \vee \texttt{false} \\ &= \texttt{true}\end{aligned}$$

   (b) Compute BSearch(15, L), showing all steps.

$$\begin{aligned}\texttt{BSearch}[15, L] &= \texttt{BSearch}[15, ((15,25),(35,45))] \ \texttt{since} 15 \not> 45 \\ &= \texttt{BSearch}[15, (15,25)] \ \texttt{since} 15 \not> 25 \\ &= \texttt{BSearch}[15, 15] \ \texttt{since} 15 \not> 15 \\ &= \texttt{true,} \ \texttt{since} 15 = 15\end{aligned}$$

2. (3.5 # 16) Write a recursive search function for finding an element in a binary search tree. You should use the notion of a left or right subtree of a node in your definition. (In Definition 3.8, the left and right subtrees are T_1 and T_2, respectively.) Make sure you account for empty nodes.
   ANS:
   B_1. If T is the empty tree, then $\texttt{BSTSearch}(t, T) = \texttt{false}$.
   B_2. If T is a single node r, then $\texttt{BSTSearch}(t, T) = \texttt{true}$ if and only if $t = r$.
   R. Suppose T has root r and subtrees T_1 (on the left) and T_2 (on the right). Then

$$\texttt{BSTSearch}(t, T) = \begin{cases} \texttt{true} & : t = r \\ \texttt{BSTSearch}(t, T_1) & : t < r \\ \texttt{BSTSearch}(t, T_2) & : t > r \end{cases}$$

3. (3.5 # 18) Define a NumberSquare as
   B. A single number x.
   R. A diagram

$$\begin{bmatrix} S_1 & S_1 \\ S_3 & S_4 \end{bmatrix}$$

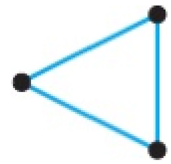
   where S_1, S_2, S_3, S_4 are NumberSquares each containing the same amount of numbers.
   Here are three examples of NumberSquares:

$$4, \begin{bmatrix} 4 & 17 \\ 13 & 1 \end{bmatrix}, \begin{bmatrix} \begin{bmatrix} 3 & 12 \\ 11 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 1 \\ 2 & 4 \end{bmatrix} \\ \begin{bmatrix} 6 & 10 \\ 7 & 3 \end{bmatrix} & \begin{bmatrix} 4 & 17 \\ 13 & 1 \end{bmatrix} \end{bmatrix}$$

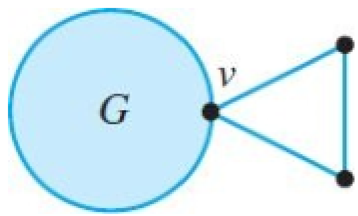
   Define a recursive function $\texttt{Trace}(S)$ that returns the sum of the upper-left/lower-right diagonal of the $\texttt{NumberSquare}\ S$. (For the examples above, the Trace function should return 4, 5, and 15, respectively.)
   ANS:
   B. Suppose $S = n$. Then $\texttt{Trace}(S) = n$.

R. Suppose $L = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \end{bmatrix}$. Then $\texttt{Trace}(S) = \texttt{Trace}(S_1) + \texttt{Trace}(S_4)$.

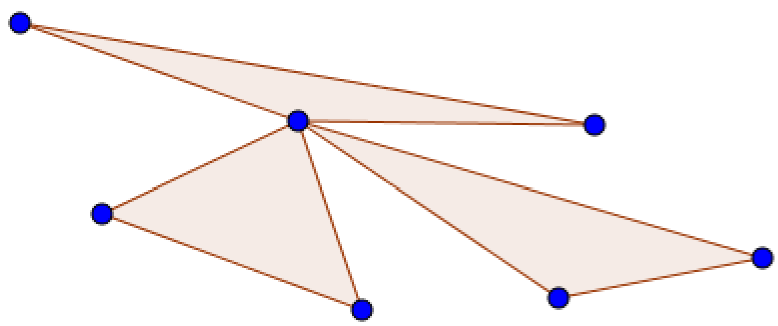
4. (3.5 # 20) Define a `TGraph` as follows.
   B. This is a `TGraph`:



   R. If G is a `TGraph` and v is a vertex of G, then this is also a `TGraph`:

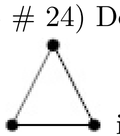


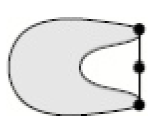
   Draw an example of a TGraph with seven vertices.
   ANS:

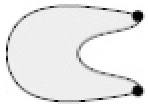
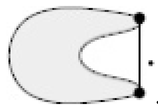


5. (3.5 # 22) Refer to Exercise 20. Prove, by induction, that any `TGraph` can be three-colored.
   ANS: Proof. (Induction on the recursive definition of a `TGraph`.) The base case of a `TGraph` contains only three vertices, so they can be colored three different colors. Suppose as inductive hypothesis that some `TGraph` G can be 3-colored. Let v be a vertex in G, and suppose a new `TGraph` is formed by adding two vertices v_2 and v_3 and connecting them to each other and to v, as in the recursive case of the definition. Let c_1 be the color of v, and let c_2 and c_3 be the other two colors in the 3-coloring of G. Assign color c_2 to v_2 and color c_3 to v_3. By inductive hypothesis, no two adjacent vertices of G share the same color, and by construction, the new vertices v_2 and v_3 are not adjacent to vertices of the same color. So the new `TGraph` can be 3-colored. By induction, any `TGraph` can be 3-colored.

6. (3.5 # 24) Define a Kgraph as follows.

   B.  is a `KGraph`.

   R_1, If  is a `Kgraph`, so is . (Any edge can be bisected.).

R_2 If  is a `Kgraph`, so is . (Any two vertices can be joined.)

Prove that in any `Kgraph`, the number of edges is greater than or equal to the number of vertices. Use induction on the recursive definition.

ANS: . *Proof.* The base case of a `KGraph` has three vertices and three edges, so $E \geq V$, where E is the number of edges and V is the number of vertices. Suppose as inductive hypothesis that a given `KGraph` has $E \geq V$. Applying part R_1 of the definition increases both E and V by 1, maintaining the inequality. Applying part R_2 of the definition increases E by 1 and leaves V unchanged, also maintaining the inequality.