CSC196: Great Ideas in Computing

Tutorial 4

28th October 2022

Assignment 2

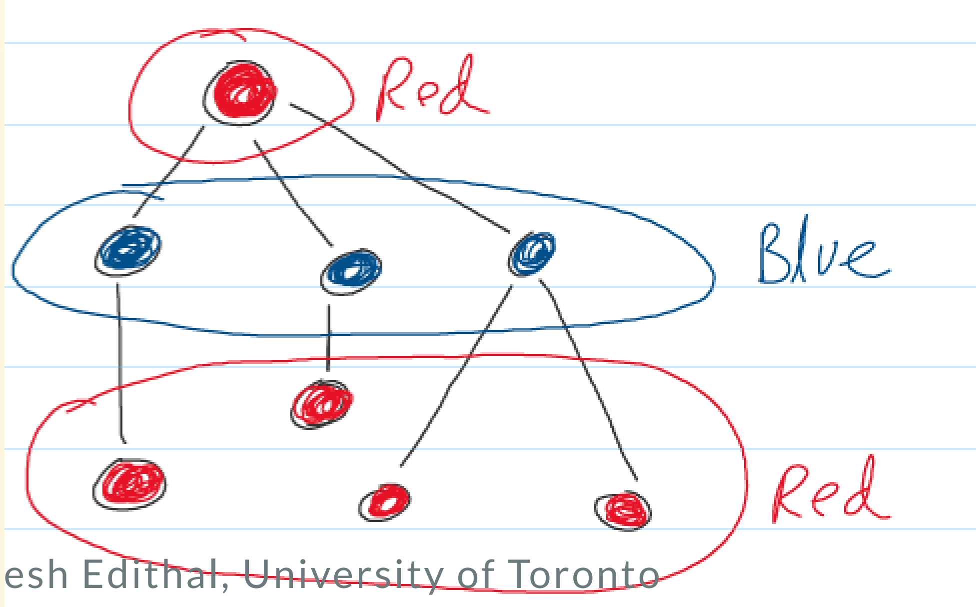
- Deadline: Wednesday, November 2, 9 AM EST
- Please try to export your solutions to .pdf before submitting, it makes grading easier
- Assignment created on MarkUs with 48 hours penalty decay of 0.5% per hour
- You have one week from the deadline to ask for re-grading
  - You will get the assignment grades within 4 days from the deadline

Q1.1

- Deciding whether a graph is k colorable is NP problem
  - Verifying whether a given k coloring of a graph is valid is P
- Naive approach (a.k.a Brute Force): Generate all colorings then check their validity
- Number of colorings: Each of the $|V|$ nodes can have k colors
- Time complexity to verify validity of coloring: For each node check their neighbors
  - Time complexity = $O(f(|V|))$ (what is the function f)
- Time complexity to decide whether graph is k colorable:
  - Number of colorings * Time complexity to verify each coloring

Q1.2

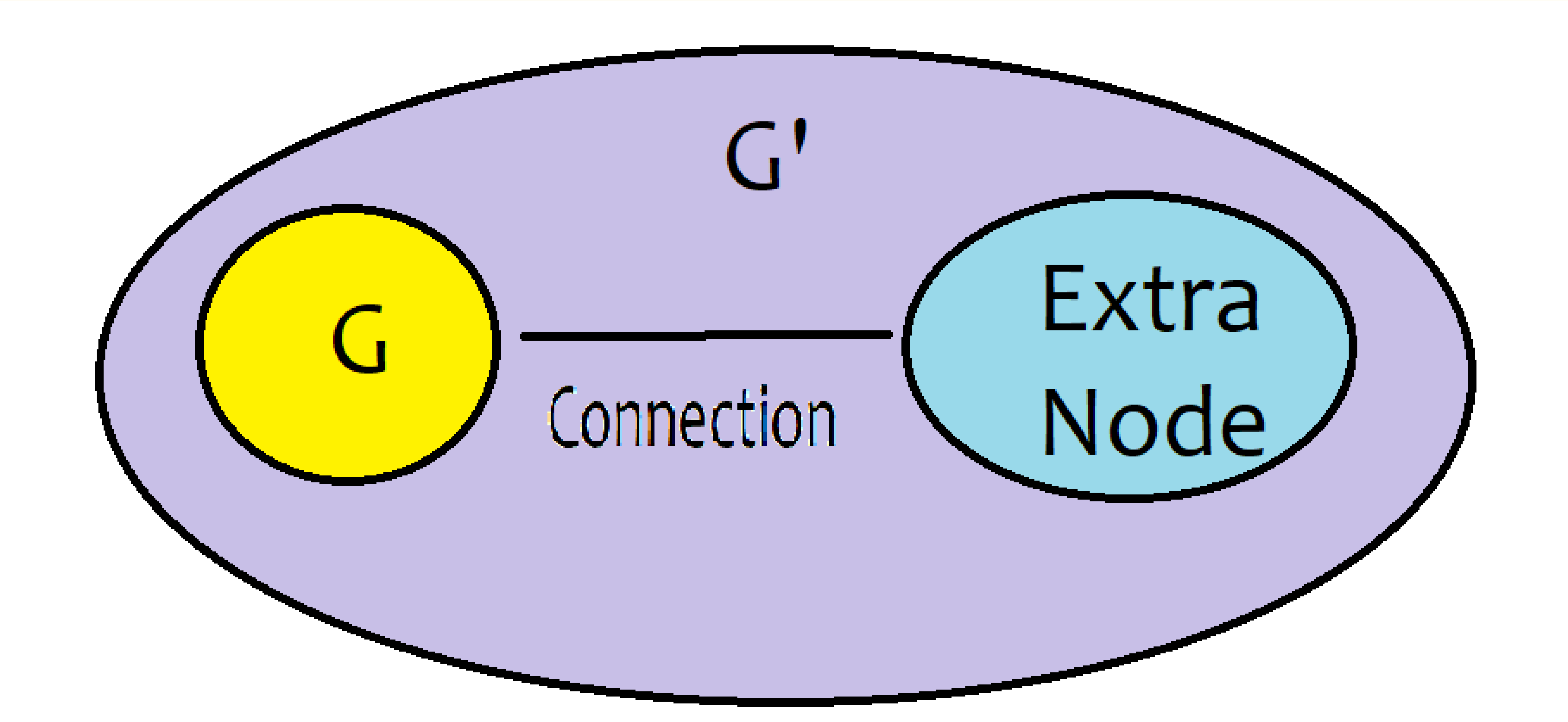
- Every tree can have a valid 2 coloring
- Choose a root node and perform breadth first search (BFS)
- Every level corresponds to nodes with the same depth from root
- Adjacent levels should have different colors



Q1.3

- Transform graph G to G' such that G is k colorable iff (if and only if) G' is $k+1$ colorable
- Transformations include adding new vertices/edges
- G' should have an extra node with completely different color than the k colors
  - How would you connect this extra node to the rest of the graph?

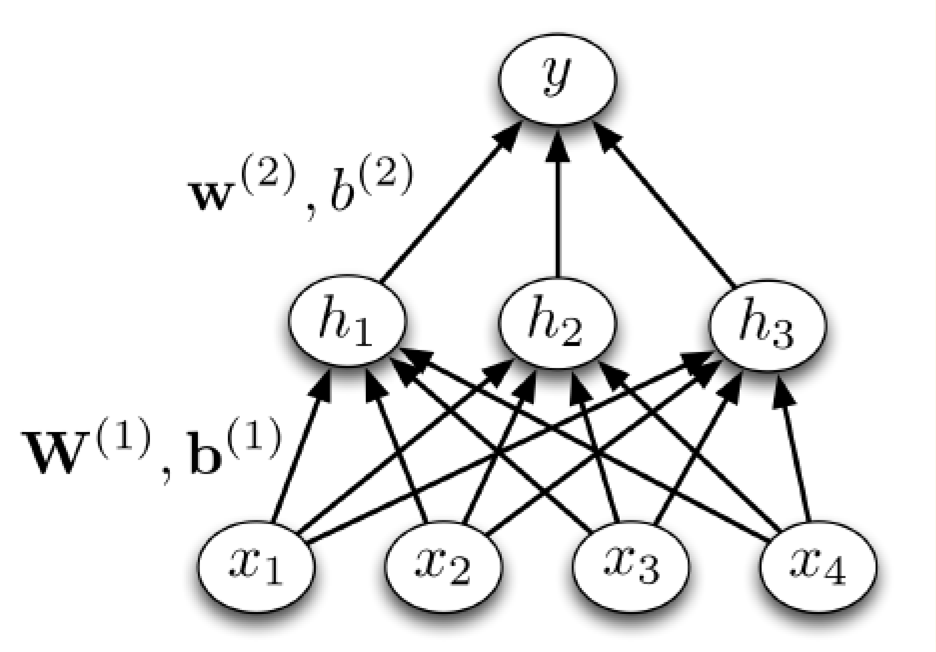
Q1.3 (continued)



Q2

- If halting problem is solved then a more powerful machine than turing machine will be possible
- Are there any ideas from Turing thesis which is not relevant to turing machines?
  - E.g. Reducibility will still hold
- This is a thought question, no best answer!
- Please justify your answer in detail

Q3



- Input layer size = 4
- Hidden layer size = 3
- Output layer size = 1

- Naming convention:
  - w_{ij}^{k} is the weight from i^{th} node of layer $k-1$ to jth node of layer k
  - b_i^k is the bias of i^{th} node in layer k
- $h_1 = f(w_{11}^2 x_1 + w_{21}^2 x_2 + w_{31}^2 x_3 + b_1^2)$, where f is non-linear step function
  - $f(x)$ is 0 for negative values of x and 1 for zero or positive values of x
- Similarly, $y = f(w_{11}^3 h_1 + w_{21}^3 h_2 + w_{31}^3 h_3 + b_1^3)$
- Number of parameters:
  - Input layer to hidden layer: (hidden_layer_size * input_layer_size) + hidden_layer_size = 4 * 3 + 3 = 15
  - Hidden layer to output layer = 1 * 3 + 1 = 4
  - Total = 15 + 4 = 19

- How would you set these 19 parameters (15 weights and 4 biases) such that y is 1 if $x_1 < x_2 < x_3 < x_4$ and 0 otherwise
- Think about what do you want h_1, h_2 and h_3 to denote
  - Possible values are $\{0, 1\}$ (boolean value)
  - Looks like the output of a logical comparison
- The objective function y is dependent on three conditions which is equal to the number of nodes in the hidden layer!

Best of luck!