Carved Write Up

**Description of Algorithm**

*Find Vertical Seam*

I initialize a node for every pixel with the node’s energy value. I set the weight to Max Value, unless the node is on the top row of the image, then I set the weight to zero. Then I iterate threw every node. I look at the nodes potential children then determine if their parent should be set to the current node I am looking at. I determine this by checking if the node’s weight plus the node’s energy is less than the child’s current energy. If it is, the child’s parent becomes the node and the child’s weight becomes the node’s weight plus the node’s energy. Then I look at the nodes equivalent to the right-most column pixels of the image and determine which node has the least weight. Once the least weight is determined I traverse through the parent’s and add them to an array that represents the path.

*Find Horizontal Seam*

I initialize a node for every pixel with the node’s energy value. I set the weight to Max Value, unless the node is on the left-most column of the image, then I set the weight to zero. Then I iterate threw every node. I look at the nodes potential children then determine if their parent should be set to the current node I am looking at. I determine this by checking if the node’s weight plus the node’s energy is less than the child’s current energy. If it is, the child’s parent becomes the node and the child’s weight becomes the node’s weight plus the node’s energy. Then I look at the nodes equivalent to the right-most column pixels of the image and determine which node has the least weight. Once the least weight is determined I traverse through the parent’s and add them to an array that represents the path.

*Remove Vertical Seam*

I determine if the array length being sent through the parameter is equal to the picture’s height or the picture width is equal to one, if so I throw an illegal argument exception. I iterate through the array and get the index. Then I iterate through the rest of the row, starting at the index plus one. I then swap the pixels. Then I cut of the right-most column from the image.

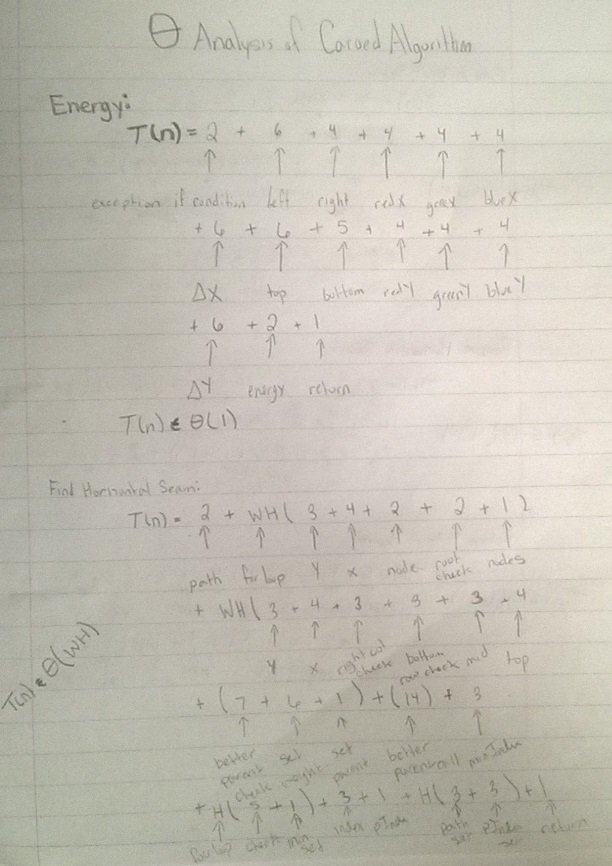
*Remove Horizontal Seam*

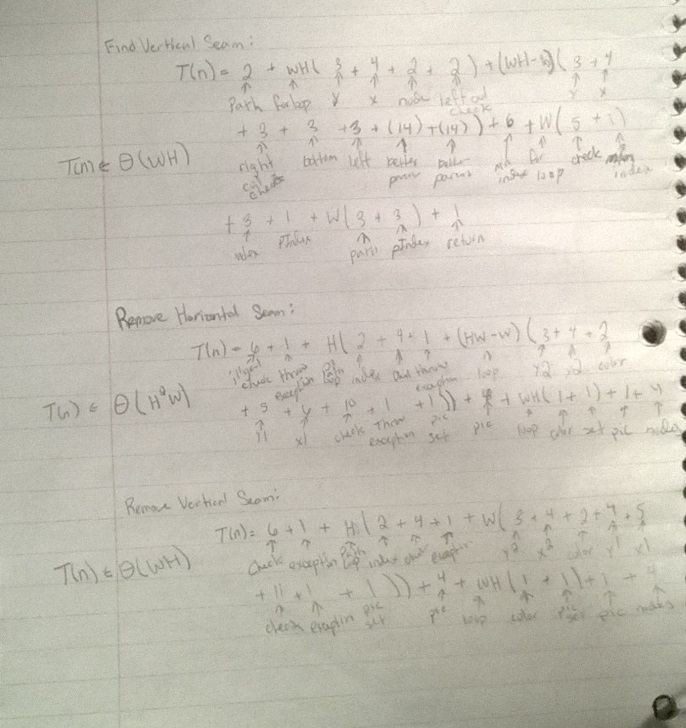
I determine if the array length being sent through the parameter is equal to the picture’s width or the picture’s height is equal to one, if so I throw an illegal argument exception. I iterate through the array and get the index. Then I iterate through the rest of the column, starting at the index plus the picture’s width, to get to the bottom pixel. I then swap the pixels. Then I cut off the bottom row from the image.

*Energy*

I check if the x and y are in bounds of the image, if not I throw and index out of bounds exception. I get the left pixel’s color and the right pixels color. I calculate red, green and blue by subtracting the right color from the left color. The change in X color is red, green and blue squared then added together. I get the top pixel’s color and the bottom pixel’s color. I set the red green and blue by subtracting the bottom from the top. The change in Y is calculated by squaring the red, green and blue then adding them together. The total energy is the change in X + the change in Y.

**Theta analysis of algorithm**





**Help**

No help.