Theoretical questions:

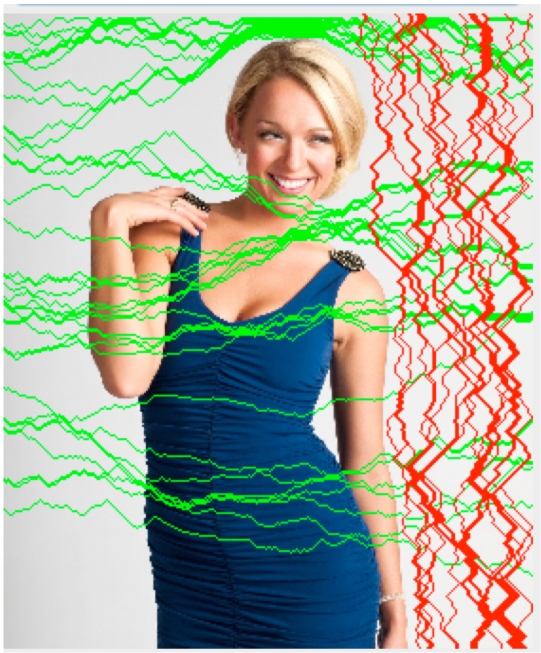
1.

a. In our implementation when we make a reduce by n1 seams and then by n2 seams will be the same as reducing n1 +n2 seams at once, because we actually plan the reduction of the seams as one by one.
Every time that we find a seam that should be removed we calculate the cost matrix again in order to find the next seam.
so it doesn't matter if we reduce the n1 and then the n2 or we reduce n1 + n2 at once.

b. . In our implementation when we make enlarge by n1 seams and then by n2 seams will be the same as enlarging n1 +n2 seams at once, because we actually plan the enlargement of the seams as one by one.
Every time that we find a seam that should be multiplied we calculate the cost matrix again in order to find the next seam.
so it doesn't matter if we enlarge the n1 and then the n2 or we enlarge n1 + n2 at once.

c.In our implementation there will be a difference between changing the width and than the height and between changing the height and than the width.
Because when we make a change on the width we get a **new image**(without the seams that were remove) and than we change the **new image** height.(when changing the new image height we calculate the cost matrix again without the seams that were removed when we changed the width).
When we change first the height and then the width it will be the opposite.
That’s why there will be a difference between the two options.

2. There is one orientation that has continuous seams and the other does not because
In our implementation we first retarget the width, that means that we change the width and we are returning a new image, and then we are changing the new image height.
that’s why one dimension will have holes.



3. According to our implementation the same seams will have holes.
The case no is different because when we make the change on the enlargement first.
The second orientation will have more pixels in the picture that it will retarget.
and that is why the seams of it will be longer.