Hole Filling – Answers – Omer Sofer:

1. In case of m boundary pixels and n pixels inside the hole, assuming B and H were already found, the complexity of the algorithm that fills the hole is:
   - We go through n hole pixels, and for every hole pixel we do m $O(1)$ calculations, so the algorithm takes $O(m * n)$ operations.
   - Because every hole pixel can have up to 8 boundary connected pixels (by definition), we get: $m = O(8n) = O(n)$.
   - Therefore, the total complexity is: $O(n^2)$.
2. In case that we want to approximate the result in $O(n)$, assuming that we don't know anything about the hole's shape, we will have to go through all the n hole pixels and make $O(1)$ calculations for each one of them → we can't take into account all m boundary pixel for each hole pixel.
   Therefore, in order to approximate the result, we will define a <u>constant</u> M. For every hole pixel hp we will look for boundary pixel in a $M * M$ square around hp, and take only those boundary pixels into account when calculating $I(hp)$ according to the given algorithm.
   - The idea behind this method is that far pixels will have lower (and maybe neglectable) weight, and therefore taking only constant number of closer boundary pixel may be good enough.
   - In the given square we will have up to $M * M$ boundary pixels, and because M is constant - we will have $O(1)$ such boundary pixels for each hole pixel.
   - With the same complexity analysis as described in question #1, in this case we will get total complexity of $O(n)$.
   - Note (without proof): if we go through the hole pixels in the order we found them – each pixel will have some boundary pixels to take into account in the $M * M$ square at the time we get to calculate it's color (including already filled hole pixels).
   - I believe that this method will be affected from the order we scan the hole pixels and maybe won't give the highest degree of accuracy, but this is the best I could think about until now. ☺