

pseudo code growing algorithm

input:

- indices of the starting points of the functions (placed according to the graph relaxation)
- integer x, y and z coordinates of the pointcloud left over after the solar envelope
- noise- (and shadow-) values for each point in the pointcloud

algorithm:

- define the characteristics of each function in the building (maximum surface, maximum shadow value and minimum noise value)
- create a panda dataframe with columns for:
 - the x-, y-, z-coordinates
 - the noise value of each point in the pointcloud
 - the shadow value of each point in the pointcloud
 - label of the function that the point belongs to (-1 = no function, to start with)
- for all the starting points: set the label in the 'function' column to the corresponding number (Atrium = 0, student housing = 1, assisted housing = 2, ...)
- set the labels of the voxels on top of the atrium starting point to 0 too, as the atrium should reach to the top of the building + the labels of the voxels just on top of the public function starting points to their corresponding labels, as all the public functions should be double height
- while loop that loops through every growing function until there are no more voxels left over in the building OR until all the maximum surfaces of the functions have been reached
 - following loop is written for each function in the building:
 - if statement that checks if a function hasn't reached its full surface yet
 - for every voxel currently labeled as 'function':
 - find the neighbouring voxels in x, y and z direction
 - candidates = neighbours that don't have a function yet
 - accepted candidates = candidates that fulfill the noise value requirement of the function
 - closest candidate = select the voxel from the accepted candidates that is closest to a neighbouring function / sunlit area/...
 - atrium: closest to the other side of the building + the park (so the atrium will have a better chance at connecting all the functions of the building)
 - public functions: closest to the atrium, so that all the functions will grow around atrium, giving the best possible network in the building

- workspaces: closest to the edge of the building, so that the work spaces will receive enough daylight
- change the label of the closest candidate in the functions column to the right number, corresponding to it's new function
- if applicable (atrium + public functions), also set the function label(s) of the voxel(s) on top of the closest candidate to the new number
 - if statement that checks if there were any accepted candidates found, if not: print a message that tells you that the growth of this function was interrupted & tell python that this functions is done with growing (functionFull = 1)
 - if statement that tells python that this function is done with growing if the maximum surface has been reached (functionFull = 1)
 - at the end of the while loop: create a counter that checks if there a still voxels left without a function so that there is still the ability to grow
- check how tall the functions have grown around the atrium and take away the voxels of the atrium above that certain level
- create lists of the index numbers of the voxels in each function

output:

- lists with indices of the voxels in each function

