4.1 Give the name of the algorithm that results from each of the following special cases:

a. Local beam search with k = 1.

If k = 1 in a local beam search then the search only keeps track of a single state and begins with 1 state and then generates 1 successor of that state, selects it and continues the search. This is analogous to random-restart hill climbing.

b. Local beam search with one initial state and no limit on the number of states retained.

If local beam search had one initial state and retained k = states then it would also generate k = a states i.e. all the children of that state assuming a < and then choose k = states to put into a list and then chooses the best one based on some heuristic to begin the process again on so this is pretty much a breadth-first search with loop detection.

c. Simulated annealing with T = 0 at all times (and omitting the termination test).

If T is 0 at all times then simulated annealing results in probability calculations = = = = 1 then the algorithm would reject any downward successor with probability 1 => first choice hill climbing

d. Simulated annealing with T = at all times.

If we took T = then the probability calculation would be probability = = = = = which implies the search would select any node => depth first search

e. Genetic algorithm with population size N = 1.

If the population size was one for a genetic algorithm then the first iteration would consist of a child being bred from two of the same parents and then, whenever the probability was right, the child (which would simply be a duplicate of the parent) would be mutated, changing a small feature about it. This child would then be placed back into the pool and would be bred again with the parent or possibly with itself depending on its fitness, which would result in duplicates of the child or the parent, implying that the only movement towards a goal state that this algorithm would have would be during random mutations thereby nullifying the advantages gained from the crossover function and turning generic algorithms into a random search.