1. Greedy -- If we consider using a greedy algorithm, we will use a heuristic that evaluates the benefits of conquering each neighboring kingdom Ki to the current kingdom Kj we're in. The heuristic is determined by the travelling distance from K to Ki, the conquering time of Ki, and a utility function that represents the benefit of conquering Ki made up of all conquering costs of Ki's neighboring kingdoms which is Ui.

   H = Ui – cii – cij

   This is because we can save the conquering cost of those neighboring kingdoms by conquering Ki only.

2. Approximation Algorithm -- We have come to a conclusion that this problem is a combination of Minimal Dominating Set and Minimal Spanning Tree problem. Because conquering a kingdom creates a set of kingdoms that surrender to us, we can spend as less time as possible in conquering kingdoms by taking the optimal Minimal Dominating Set and minimizing the cost of the optimal Minimal Dominating Set. Second, in order to find the best conquering choice of each kingdom, we first need to decide an optimal Since this problem considers incorporates finding a minimal tour starting and ending in the same vertex, we also have to solve the minimal route problem.

   First, we have to solve the Minimal Dominating Set problem by assigning each kingdom with a set that covers itself and all its neighboring kingdoms. Each set Si has a value of the conquering time of the center kingdom cii.

   Then, we define the optimal Minimal Dominating Set to be able to cover all kingdoms while also having the smallest sum of Si's.

Next, we find a minimum spanning tree that starts at our first kingdom, connects all the center kingdoms in the optimal Minimal Dominating Set, and returns back to the starting kingdom. To solve this, we have to design an approximation of the Travelling Salesman Problem which does not limit that each vertex can be travelled only once.