1. **What are the criteria for a recursive solution?**

* It should be possible to reduce the size of the problem, repetitively.
* The endpoints of the solution (base case) should at one time be met (so that StackOverFlow does not occur), therefore the base case should be identified and organized to fit the problem.
* There must also be a way of getting a larger result from each of the divided component elements.

1. **Think of another everyday example. (Of recursion)**

One set of everyday examples of recursion is fractals in real life. Fractals are patterns that are for the most part, never ending, they are created by repeating an initial pattern in a feedback loop. They are driven by recursion as each instance of the pattern is a smaller/or larger version of the last subsequent one. In nature, this pattern isn't always never ending, but it exhibits the same properties. Some of the best examples would be: Onions, Romanesco broccoli, cabbage, trees, frost crystals, and a lot more.

1. **Give two specific problems associated with recursion, and two reasons for using recursion.**

**Problems-**

Recursion algorithms are for the most part slower than their iterative counterparts because of the overhead of method calls. Recursion methods also have higher space complexities as they keep stacking up in memory.

**Reasons For-**

Recursion solutions are usually simpler,shorter and easier to understand than other forms of solving a problem (iterative). It also reflects the abstract solution strategy (meaning its quality is on pare to standards set by the IEEE), as many algorithms are also defined recursively it's easy to implement them.

1. **What does an iterative solution mean?**

An iterative solution is a method of solving a problem that includes the repetitions or iterations of a set of instructions a set amount of times until a certain condition is met. When the first set of instructions is re-executed, it is called an iteration.