Memoization

Here is a nice example of a C# memoization function to speed up calculations.

```
public static class Memoization
{
    public static Func<T, TResult> Memoize<T, TResult>(this Func<T, TResult> function)
    {
        var cache = new Dictionary<T, TResult>();
        var nullCache = default(TResult);
        var isNullCacheSet = false;
        return parameter =>
        {
            TResult value;

            if (parameter == null && isNullCacheSet)
            {
                return nullCache;
            }

            if (parameter == null)
            {
                nullCache = function(parameter);
                isNullCacheSet = true;
                return nullCache;
            }

            if (cache.TryGetValue(parameter, out value))
            {
                return value;
            }

            value = function(parameter);
            cache.Add(parameter, value);
            return value;
        };
    }
}
```

Unit test

```
[TestFixture]
public class MemoizationTest
{
    [Test]
    public void MemoizeFunctionWithOneArgument()
    {
        Func<double, double> sinFunc = System.Math.Sin;
        var sin = sinFunc.Memoize();

        // Build sin(x) lookup table
        for (int i = 0; i < 100; i++)
        {
            sin(i);
        }

        Assert.AreEqual(-0.99975517335861985, sin(55), 0.001);
    }
}
```