CSC 260, Spring 1999 - Assignment #1

Due at start of tutorial on February 12. Worth 10% of the course grade.

Note that this assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own.

In this assignment, you will investigate using a Taylor series to compute the binary entropy function, $H(p) = -p\log(p) - (1-p)\log(1-p)$. This function is defined for p between 0 and 1, and is zero for $p = 0$ or $p = 1$. It gives the amount of "information" contained in a single binary variable that has the value 1 with probability p and the value 0 with probability $1-p$. It is used extensively in information theory and in applications such as data compression and error-correcting codes. The base of the logarithms used in the definition is often taken to be two, but here we will use natural (base e) logarithms.

Of course, you can compute $H(p)$ from the formula given above, as long as you can compute logarithms (though the cases $p = 0$ and $p = 1$ may need to be done specially). Computing $H(p)$ this way requires computing two logarithms, however. We might hope that computing $H(p)$ directly would be faster. It's a good exercise in any case.

Here is a Maple procedure to compute H from its definition, followed by a use of it to plot the function:

```
> H := proc (p) -p*log(p) - (1-p)*log(1-p) end;
> plot(H(p),p=0..1,numpoints=1000);
```

You should enter these commands to see what the function is like, and so you can use `H` below.

Question 1: Use Maple's symbolic differentiation capabilities to help you to figure out the Taylor series for $H(p)$ about $p = 1/2$.

First, you should find the first few deriviatives of H with commands such as

```
> diff(H(p),p);
> diff(H(p),p$2);
> diff(H(p),p$3);
  etc.
```

From the output of these commands, make a guess at the general pattern that the derivatives follow, and verify that guess by hand, using mathematical induction.

To find the Taylor series for $H(p)$ about $p = 1/2$, you will need the general formula for the n'th derivative evaluated at $p = 1/2$, which you should be able to get once you know the pattern of the derivatives. Plugging this into the general form of the Taylor series should give you the general pattern that the series follows.

You should verify that you have done this correctly by checking your general description of the Taylor series against the first few terms of the Taylor series as found by Maple, with the following command:

```
> taylor(H(p),p=1/2);
```

Finally, you should consider whether or not this series will converge for all p in the range from 0 to 1. You're not required to prove anything about this for this assignment, however.

Question 2: Write a Maple procedure called `H_taylor` to compute a floating-point approximation to $H(p)$ by summing the Taylor series about $p = 1/2$ to saturation (ie, until adding the next term doesn't change anything, given the precision of the floating-point arithmetic being used).

Try to make your procedure as efficient as possible. In particular, when computing the next term of the series, try to make use of things that were computed for the previous term. You should be able to compute each term using a constant number of arithmetic operations, that doesn't grow bigger for later terms.

You may need to compute some constant (not depending on p) as part of the series. If you were really trying to save computing time, you would compute this constant just once, not over and over again for every new value of p, but for this assignment, computing it every time is OK, since that will be a bit more convenient.

The `H_taylor` procedure should return the approximation to $H(p)$ as its value. It should also set the global variable `terms_used` to the number of terms in the series that were added together to find this approximation (don't count any terms that were left out because they were known ahead of time to be zero).

Hand in a few tests of this procedure that confirm that it generally works for values of p that are not close to 0 or 1. You should compare the results you get with those obtained using the `H` procedure defined above, but the results could be slightly different due to the effects of round-off error.

Question 3: Test your `H_taylor` procedure to see how accurate it is, and how much time it takes, as measured by how many terms of the series it needs to use.

Both the accuracy and the efficiency will vary with the precision of the floating-point arithmetic used, which is determined by setting the `Digits` variable. For your initial tests, set `Digits` to 5, but you may want to do tests with it set higher as well.

To judge the accuracy of `H_taylor`, you will need to know what the correct answer is. You can get an idea of this using the `H` procedure defined above, and also by using both `H` and `H_taylor` with a larger setting of `Digits`.

Pay particular attention to values of p that are close to or equal to 0 and 1, since it is in the vicinity of such extreme points that you might expect the biggest problems.

You should conclude by summarizing how well this method of computing $H(p)$ works, as determined from your tests.