1. Explain briefly your background and what made you interested in this course. What is your major subject (computer science, math, ...), and at what stage of your studies you are? How much mathematics have you studied (name some courses you have taken)? How about programming—what language(s) you use, and how would you describe your programming skills? If you have studied something else relevant, like statistics, signal processing, etc., you can also mention that. Have you studied any information theory before?

2. Find out what compression algorithms are used in the gadgets you use—DVD player, digital TV, CD, iPod, cell-phone, laptop (WinZip), etc. You can simply list the names of the algorithms, you don’t have to study how these algorithms work.

Solution  Compression algorithms split into two categories, lossy and lossless compressors. Lossy encoding is widely used for signal processing, e.g. mp3, MPEG-4, JPEG and the like. Lossless compression is needed for binary data, e.g. gzip, bzip2, PKZIP and so on.

3. In a programming language of your choice, find as short a program as you can for printing out the sequence 141592653589793...375678 (the first 10000 digits of π after the decimal point). Search the web to find an algorithm for calculating π. Your solution to the problem should contain your source code and a brief description of the algorithm you use (the formula and where it’s from).

Solution  For instance, we have

\[ \pi = 3 - 4 \sum_{i=1}^{\infty} (-1)^i (i(i+1)(i+2))^{-1}. \]

This is the Nilakantha series and can be found at Wikipedia. The series of n-term finite sums alternates around the correct value of π, such that the nth term is an upper bound to the remaining error. Using some library that allows sufficient precision, a suitable program may look like this:

```cpp
pi=3;
eps=pow(10,-10000);
term=1.0;
i=1;
while(term > eps){
    term = -4*pow(-1,i)/(i*(i+1)*(i+2));
    pi += term;
    i++;
}
output(pi, 10000);
```

This is not the fastest algorithm, but the goal was a short code, running time is not an issue here!

4. For 0 ≤ p ≤ 1, we define the binary entropy \( H(p) \) as

\[ H(p) = -p \log_2 p - (1 - p) \log_2 (1 - p). \]
This is the entropy of a random variable that gets value 0 with probability $p$ and value 1 with probability $1 - p$, and will appear frequently during the rest of the course.

Prove that $H(p)$ is strictly concave and achieves its maximum at $p = 1/2$. (Hint: Remember that a sufficient condition for concavity is that the second derivative is negative.)

Solution

$$H(p) = -p \log_2 p - (1 - p) \log_2 (1 - p) = -\log_2 e(p \ln p + (1 - p) \ln(1 - p))$$

$$H'(p) = -\log_2 e(\ln p + 1 - \ln(1 - p) - 1) = -\log_2 e(\ln p - \ln(1 - p))$$

$$H''(p) = 0 \iff p = \frac{1}{2}$$

$$H''(p) = -\log_2 e(\frac{1}{p} + \frac{1}{1 - p}) < 0$$

5. We write $\mu = E[X]$ for the expectation and $\sigma^2 = \text{Var}[X]$ for the variance of a random variable $X$. (Recall that $\sigma^2 = E[(X - \mu)^2]$.)

(a) Prove Markov’s Inequality: for any non-negative random variable $X$ and any $\delta > 0$, we have

$$P(X \geq \delta) \leq \frac{\mu}{\delta}$$

Solution Define the indicator variable

$$I_{X \geq \delta} = \begin{cases} 1 & \text{if } X \geq \delta \\ 0 & \text{else} \end{cases}$$

Then

$$\delta P(X \geq \delta) = \delta E[I_{X \geq \delta}] \leq E[X] = \mu$$

provides the required.

(b) Apply Markov’s Inequality to the random variable $(X - \mu)^2$ to obtain Chebyshev’s Inequality: for any $\delta > 0$ we have

$$P(|X - \mu| \geq \delta) \leq \frac{\sigma^2}{\delta^2}.$$ 

Solution

$$P(|X - \mu| \geq \delta) = P((X - \mu)^2 \geq \delta^2) \leq \frac{E[(X - \mu)^2]}{\delta^2} = \frac{\sigma^2}{\delta^2}.$$

(c) Consider now $X$ that follows binomial distribution with parameters $n$ and $p$. That is, $X$ is the number of heads in $n$ independent tosses of a biased coin with probability $p$ of getting heads. We know from a basic course in probability that now $\mu = np$ and $\sigma^2 = np(1 - p)$. Use Chebyshev’s Inequality to give an upper bound for the probability that the value of $X$ differs from its expectation by at least 5%; that is, $|X - \mu|/\mu \geq 0.05$. Give a general formula and calculate numerical values for $p = 0.5$ and $p = 0.8$, with $n = 20$, $n = 100$ and $n = 1000$.

Warning: This gives very loose upper bounds. In practice you should just calculate the values exactly (if you need specific numerical values) or use Chernoff bounds or similar (if you need an asymptotically fairly accurate closed-form formula for theoretical purposes); see the Bonus Problem.
Solution

\[ P(|X - \mu|/\mu \geq 0.05) = P(|X - \mu| \geq 0.05\mu) \leq \frac{np(1-p)}{0.05^2 \mu^2} = \frac{1-p}{0.0025np} \]

\[
\begin{array}{c|ccc}
\frac{1-p}{0.0025np} & n = 20 & n = 100 & n = 1000 \\
p = 0.5 & 20 & 4 & 0.4 \\
p = 0.8 & 5 & 1 & 0.1 \\
\end{array}
\]

**Bonus Problem** Consider a binomially distributed random variable \( X \) with parameters \( n \) and \( p \), as in Problem 5(c).

(a) We know that for \( k \in \{0, \ldots, n\} \) we have

\[ P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}. \]

Use this to calculate the probability for \( X \) differing from its expectation by at least 5%, for the same combinations of \( n \) and \( p \) as in 5(c). That is, you should calculate the actual values for the probabilities you upper bounded in 5(c) using Chebyshev’s Inequality.

**Solution**

\[
\begin{array}{c|ccc}
P(|X - \mu|/\mu \geq 0.05) & n = 20 & n = 100 & n = 1000 \\
p = 0.5 & 0.824 & 0.617 & 0.121 \\
p = 0.8 & 0.782 & 0.381 & 0.002 \\
\end{array}
\]

(b) Compare the probabilities you calculated in part (a) with upper bounds you get from the Chernoff bound

\[ P(|X - \mu| \geq \delta \mu) \leq 2 \exp(-\mu \delta^2/3). \]

**Solution**

\[
\begin{array}{c|ccc}
P(|X - \mu| \geq 0.05\mu) & n = 20 & n = 100 & n = 1000 \\
p = 0.5 & 1.98 & 1.92 & 1.32 \\
p = 0.8 & 1.92 & 1.87 & 1.03 \\
\end{array}
\]

All of these bounds are useless, as probabilities are bounded by 1 anyway.