Middle East Technical University
Department of Computer Engineering

CENG 140

Spring 2017

Take Home Exam 1

REGULATIONS

Due date: 10 May, 2017 Wednesday *(Not subject to postpone)*

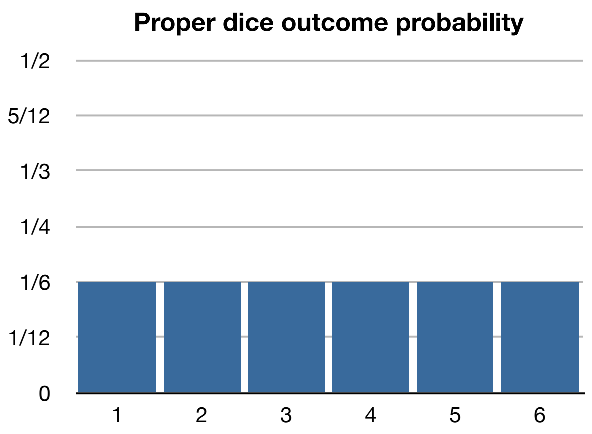
Submission: Electronically. You will be submitting your program source code through a text file which you will name as `the1.c` by means of the COW system.

Team: There is **no** teaming up. The homework has to be done and turned in individually.

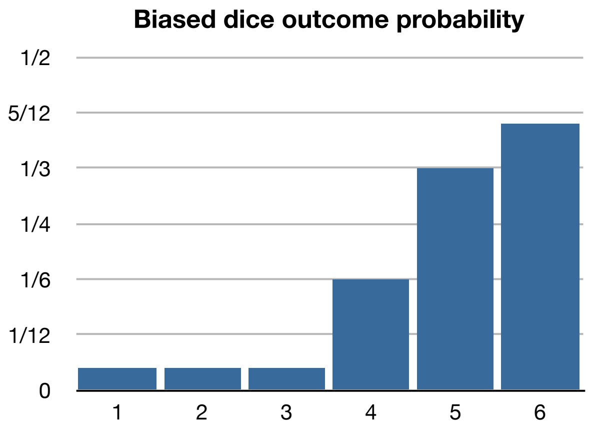
Cheating: Source(s) and Receiver(s) will receive zero and be subject to disciplinary action.

INTRODUCTION

A *stochastic variable* or *probabilistic variable* is a variable which value is non deterministic. For example a ⟨dice outcome⟩ can be a part of a formula (probably a game theoretical one) as a stochastic variable. As you would agree, the variable will have integer values in the range [1,6] with equal probabilities of 1/6. So if we would plot the function of the outcome value versus probability we would have:

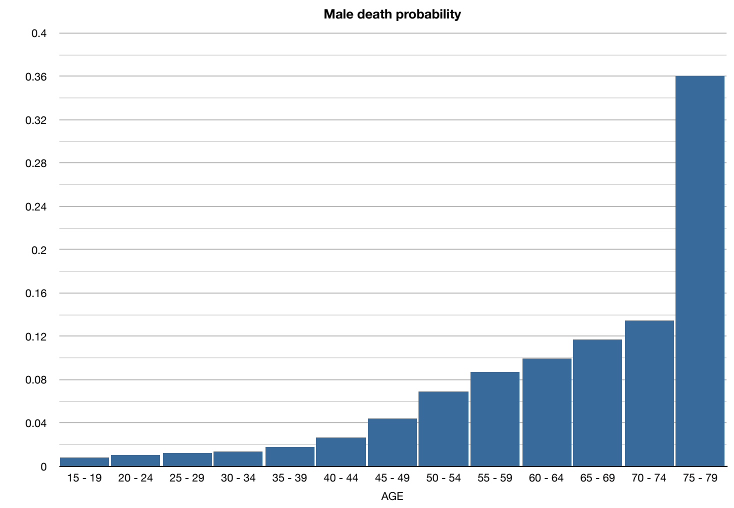


This is ofcourse so only if the dice is unbiased. If it is not so, then the evenness in the function would disappear. And we would probably have something like:



The sum of all probabilities though would still be 1 (unity). In other words the probabilistic function is normalized to 1. These functions define the stochastic variable ⟨ dice outcome⟩ as a set of possible values each of which occurs with a certain probability (defined by the function).

Many probabilities are depending of the value of the outcome. For example, based on demographic work, if a male person is going to die in the age range of [15,79] then the probability varying with the age:



Certainly, the increase in the functions of the examples are pure coincidence. The function can have any shape: for example the shape of the normal distribution, which is found in many natural and social events is of a shape of a bell.

Now what if this type of variable participate a function? What can we say about the result of the function? It will be some probability distribution function. Depending on the possible values the stochastic (ingredient) variables take the function will take some values. Depending on the probability of those possible values the probability of the outcome will shape.

PROBLEM

How can this probability distribution function be calculated? This is the task of this take home exam.

If we have the chance to perform repetitive experimentation of considerable amount then we can determine the distribution by performing random generated experimentation and calculate the outcome for the function. We will do so by 'drawing' values for each variable with probabilities set by their associated probability distribution functions. Then we substitute these values for the variables and do a calculation of the formula that defines the function. We record the resulting value. Repeating over and over again this task will give us a histogram of the outcome values. The last task is to convert the histogram into a probability distribution by normalizing it.

In the input you will be given an infix formula that defines the function we are going to compute the probability distribution function of. Furthermore, for each variable in the formula you will be given an ordered set of n tuples. Each tuple is a value and the probability that value will occur. The probabilities will be normalized (ie. they add up to unity).

Here is the BNF representation for the formula:

$$\begin{array}{rcl}
\langle\text{formula}\rangle & ::= & \langle\text{formula}\rangle\,\langle\text{operator}\rangle\,\langle\text{formula}\rangle \mid \\
 & & \texttt{(}\,\langle\text{formula}\rangle\,\texttt{)} \mid \\
 & & \langle\text{function}\rangle\,\texttt{(}\,\langle\text{formula}\rangle\,\texttt{)} \mid \\
 & & \langle\text{letter}\rangle \mid \langle\text{unsigned number}\rangle \\
\langle\text{operator}\rangle & ::= & \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \texttt{/} \mid \texttt{\^{}} \\
\langle\text{function}\rangle & ::= & \texttt{\~{}} \mid \texttt{sin} \mid \texttt{cos} \mid \texttt{sqrt} \mid \texttt{ln} \\
\langle\text{letter}\rangle & ::= & \texttt{A} \mid \texttt{B} \mid \ldots \mid \texttt{Z} \\
\langle\text{unsigned number}\rangle & ::= & \langle\text{any unsigned \texttt{\%lf} readable number}\rangle
\end{array}$$

As far as semantic is concerned:

`+`, `-`, `*`, `/`, have conventional meaning, associativity and precedence (as of C). `^` is exponentiation, has right associativity and top precedence among all ⟨operator⟩s. For simplicity unary minus, represented by `~` and subtraction have different symbols. Furthermore, again for simplicity, unary minus is a ⟨function⟩ (is followed by a compulsory pair of parenthesis). Note that we do not have direct negative number input (like `-425.`).

SPECIFICATIONS

- The ⟨formula⟩ will be the first line of the input and will not extend over the following lines. Any subpart of the input that is also a ⟨formula⟩ may be (pre/pro)ceeded by blank(s). Your implementation must be insensitive to them.
- The maximum length of the formula line is 200 characters.
- The following line contains two parameters (separated by blank(s)):
  - The count of intervals n for the probability distribution functions (also valid for the result). This is an integer in the range $[5, 100]$.
  - Count of experiments. This is an `long int`. No further limitation is given on this.
- Each of the lines starting with the third line and following it, is of the structure:
  - a ⟨letter⟩, followed by blank(s).
  - Two floating point values, the lower limit of the value axis (the x-axis) of the distribution function for the ⟨letter⟩ variable and the upper limit, respectively.
  - n probabilities. The first is the probability corresponding to the first interval, the second is the is the probability corresponding to the second interval, and so on.

  In this manner the probability distributions for all the variables will be provided. As observed, the intervals are of equal length.

  Furthermore, the probability among a single interval is constant. So, for example let us have an interval $[50, 59]$ a probability assigned that is 0.015 . This says

  $$\frac{\langle \text{count of values in the range [50,59]} \rangle}{\langle \text{count of all values} \rangle} = 0.015$$

  and the probability of having, for example, $50, 50.1, 53, 58.99$ or 59 are all the same (attn: of course each of those individual probabilities are not 0.015).
- The output is a line with a content similar to a line describing a variable. But this time we do not have the letter. The output starts with the lower and upper limit floating points, followed by n probability values. This is the computed probability distribution for the formula input. All output goes to a single line.
- There will be no erroneous input.

Here is an input example:

```
sqrt(2 - sin(3*A/B)^2.5)   + 0.5*(C*~(D) + 3.11    +B)
20 100000
C 3.25 18. 0.01 .01 .02 .04 .08 .02 .02 .05 .065 .08 .1 .13 .2 .05 .04 .04 .03 .01 .005  .0
A 0 7.5 .054 .031 .016 .008 .116 .124 .147 .155 .039 .023 .016 .008 .124 .062 .031 .016 .008 .008 .008 .006
D -1.5 0.5 .012 .025 .05 .1 .1 .1 .025 .012 0 0 0 .012 .025 .1 .2 .1 .05 .039 .025 .025
B 1 3 .117 .058 .029 .015 .007 .007 .007 .015 .022 .029 .036 .044 .051 .058 .066 .073 .080 .088 .095 .103
```

HINTS & HOW-TO-DO'S

- In your CENG 111 course you extensively studied Dijkstra's Shunting Yard Algorithm[1]. As you recall this algorithm was used to evaluate infix expressions. In this THE you are going to implement and use it. Your first task should be to discover how the Dijkstra's algorithm and the evaluation phase is going to be altered to handle ⟨function⟩. (There is a simple, though elegant solution to this).
- Write a draw function which takes as parameter your internal representation of a probability distribution and generates random values conforming with the probability distribution. You have the `random()` function available in `stdlib.h` to hand (`man random` is your friend). You have to figure out a way to get a randomization proportional to the probability values. Imagine a blind man making random dots with his pencil on a ruler...
- You do not have the chance to know what the lower and upper bounds of the outcome is. So you do not have a way to know what the intervals are. You can determine these only after the completion of all the experiments. Keeping all the data is not a good idea. So what to do? Think about it. (The evaluation process will take a reasonable tolerance into account ;-)).

Bİ GERGEDANLA AYNI ISSIZ ADAYA DÜŞME OLASILIĞIN NEDİR?
TRİLYONDA BİR FİLANDIR HERHALDE...
PEKİ O GERGEDANIN MANYAK OLMA OLASILIĞI ?..

[1]Ucoluk and Kalkan, Introduction to Programming Concepts with Case Studies in Python, Springer, 2012