CSC 284/484 - homework 1 (basic probability)

`http://www.cs.rochester.edu/~stefanko/Teaching/17CS484`

Students that take the course as 484 are required to do **both** 284/484 and 484 parts of the homework. Students that take the course as 284 are only required to do 284/484 part of the homework (of course you are welcome to solve/turn-in the 484 part as well).

---

1 284/484 homework - solve and turn in

1.1 Theoretical part

Exercise 1.1 (**due 1/31/2017**) We throw m balls into n bins (each ball is thrown into a uniformly random bin independently of the other balls). What is the expected value of the number of empty bins?

Exercise 1.2 (**due 1/31/2017**) The randomized k-select algorithm (`Select`) is similar to the quicksort algorithm. The call to `Select(A,s,t,k)` returns the element of rank `k` in `A[s...t]`. The call to `(s',t') <- Partition(A,s,t,r)` returns `s',t'` and rearranges the elements in `A[s..t]` so that the elements in `A[s...s']` are less than `r`, elements in `A[t'...t]` are larger than `r` and the elements in `A[s'+1..t'-1]` are equal to `r`.

```
Select(A,s,t,k)
  r <- A[l] where l is uniformly random from {s,...,t}
  (s',t') <- Partition(A,s,t,r)
  if k <= s'-s+1 then return Select(A,s,s',k);
  if k <= t'-s then return r;
  return Select(A,t',t,k - (t'-s));
```

Just for comparison the Quicksort algorithm is

```
Quicksort(A,s,t)
  r <- A[l] where l is uniformly random from {s,...,t}
  (s',t') <- Partition(A,s,t,r)
  Quicksort(A,s,s');
  Quicksort(A,t',t');
```

Assume that the elements contained in `A` are $a_1 < a_2 < \cdots < a_n$ (the elements do not have to occur in `A` in this order). To analyze the `Quicksort` algorithm we counted (in expectation) the number of comparisons made in the `Partition` procedure (recall that a call to `Partition(A,s,t,A[l])` compares `A[l]` with all `A[i]` for all $i \in \{s,\ldots,t\} \setminus \{l\}$). We showed, for `Quicksort`, that the probability that $a_i < a_j$ get compared is $2/(j-i+1)$. What is the probability that $a_i < a_j$ gets compared in the `Select` algorithm? (Your answer should also depend on k.) Give an (asymptotic, that is "O") upper bound on the expected number of comparisons made by the `Select` algorithm.

Exercise 1.3 (**due 1/31/2017**) What is the expected running time of the following algorithm:

$X = n$
while $X > 0$ do X = uniformly random integer in $\{0, \ldots, X-1\}$.

Asymptotic (Θ) answer is sufficient. Prove your answer.

Exercise 1.4 (**BONUS PROBLEM - due 1/31/2017**) What is the expected running time of the following algorithm for $a = 3/2$ and $b = 1$:

$X = n$
while $X > 0$ do X = uniformly random integer in $\{0, \ldots, \lfloor aX + b \rfloor\}$.

Asymptotic (Θ) answer is sufficient. Prove your answer. (Not difficult enough? How about for general $a > 0$ and $b > -a$?)

Exercise 1.5 (**due 1/31/2017**) Let p be a prime. Let n be a number $n \leq p$.

- pick random $a_0, \ldots, a_{k-1} \in \{0, \ldots, p-1\}$,
- let $X_g = a_0 + a_1 g + a_2 g^2 + \cdots + a_{k-1} g^{k-1} \mod p$ for $g = 0, \ldots, n-1$.

Prove that the random variables $X_0, \ldots, X_{n-1}$ are k-wise independent.

1.2 Applied part

Exercise 1.6 (**due 1/31/2017**) We are given n random variables $X_1, \ldots, X_n$ and we want to find the largest k such that the following statement is true: "the variables $X_1, \ldots, X_n$ are k-wise independent".

Your implementation should read the input from `stdin`. The first line of the input contains an integer $m = |\Omega|$ (the size of the sample space; w.l.o.g, $\Omega = \{1, \ldots, m\}$; each atomic event has the same probability $1/m$) and an integer n (the number of random variables). Each of the next n lines contain m integers—the i-th line contains $X_i(1), \ldots, X_i(m)$. Do not worry about efficiency of your algorithm.

Example input:

```
4 3
0 0 1 1
0 1 0 1
0 1 1 0
```

Example output:

```
2
```

2 484 homework - solve and turn in

2.1 Theoretical part

Exercise 2.1 (**due 1/31/2017**) Let $A, B \subseteq U$ be disjoint and such that $|A| = \Theta(n)$ and $|B| = \Theta(n)$. Let R be a random subset of U where each element is picked independently with probability $\Theta(1/n)$. Show

$$P(A \cap R = \emptyset \text{ and } B \cap R \neq \emptyset) > c,$$

where c is a constant (depending on the constants in the Θ's).

Exercise 2.2 (**due 1/31/2017**) Let $0 < p < 1$. Let $X_1, X_2, \dots$ be independent identically distributed random variables $P(X_i = -1) = 1 - p$, $P(X_i = 1) = p$. Let T be the smallest t such that $\sum_{i=1}^{t} X_i < 0$ (note that $P(T = 1) = 1 - p$). What is $E[T]$? (Your answer should be a function of p.)

2.2 Applied part

The objective of the next exercise is to make you familiar with the most useful construction of k-wise independent random variables (and also k-wise independent hash functions). It is the same idea as 1.5 except it uses finite fields of size 2^t. Don't be scared— You only need to understand very little about finite fields in order to implement the method:
(`https://en.wikipedia.org/wiki/Finite_field_arithmetic`).

Do implement your own finite field arithmetic (do not use other's implementations). It is fine if you use brute force to find an irreducible polynomial and generator of the multiplicative group. It is fine to have quadratic algorithm for polynomial multiplication and division. Do not optimize.

Exercise 2.3 (**due 1/31/2017**) Implement a program that generates samples from k-wise independent random variables $X_0, \dots, X_{n-1}$ with values in $GF(2^t)$. For simplicity assume $n < 2^t$. Use the following method:

- pick random $a_0, \dots, a_{k-1} \in GF(2^t)$,
- let g be a generator of the multiplicative group of $GF(2^t)$,
- let $X_i = a_0 + a_1 g^i + a_2 g^{2i} + \cdots + a_{k-1} g^{(k-1)i}$ (all computations done in $GF(2^t)$).

The input to your program is: n, k, t (where $n < 2^t$); the output is a sample from $X_0, \dots, X_{n-1}$.