CPCS 500
Lecture 6: September 25, 2013
Scribe: Fatemeh Dorri

1 Overview

In the previous lecture, Stable Marriage (proposal) problem has been introduced and an algorithm was suggested that resulted in a stable matching between two set of men and women. But the complexity of the algorithm was not discussed. In this lecture, we first review the different kinds of algorithms in terms of the randomness in either inputs or proposed algorithms. Then the complexity of different random input algorithms such as Stable Marriage problem is analyzed. The analysis on different algorithm will provide us perliminary insight to how analyze random input Algorithms.

2 Randomized/Deterministic Algorithm vs Randomized/Deterministic Inputs

The problems, depending on the nature of their input or proposed algorithms can be categorized in to four groups.

1. Deterministic input, deterministic Algorithm
   - e.g Matrix multiplication
2. Deterministic input, randomized Algorithm
   - e.g Las Vegas Algorithm, Quick sort
3. Randomized input, deterministic Algorithm
   - e.g Stable Marriage Problem
4. Randomized input, randomized Algorithm
   - e.g Quick sort on Random bits

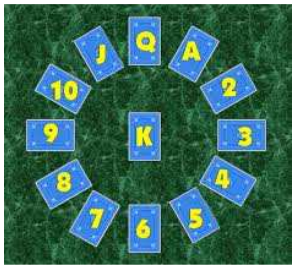
The first group has deterministic inputs and a deterministic algorithm. The result of the algorithm will not change by running it multiple times. In contrast, there could be randomness in either the inputs or algorithm or both. For example, if the input is deterministic but there is some kind of randomness in the algorithm, then the result for a certain input could be different for multiple runs of the algorithm.
Here, the focus is on the third category which the algorithm is deterministic but the inputs are random. In this case, it is desired to have an estimate of the complexity of the method like the running time of the algorithm for all different possible inputs (upper bound, lower bound, expectation or etc.).

2.1 Randomized Input – Deterministic Algorithm Analysis

We first analyze a randomized input – deterministic algorithm called “Clock Solitare”. Then we will continue to analyze the running time for Stable Marriage problem which was discussed in the previous lecture.

Clock Soliatre

It is a play with cards. In this problem, the cards are placed like a clock in thirteen piles, each having four cards, facing down. The play rule is depicted in Table 1.



There are two facts supporting this game

1. At each draw, any unseen card is equally likely.
2. Game ends with an attempt to draw from K pile.

The first fact is true because, the first card probability is uniform to be any card. The second card probability is also uniform among the remaining card and so on and so forth. The second fact is proved by contradiction:

Proof for Fact 2: The string of the revealed card can't end with something other than "K".
Assume there is an string of revealed cards that is a "WIN" but do not end with "K"

3 5 7 K ..6 8 J K…7 Q A K….3 4 K…3 4 6

As this is a WIN case, all the cards should be revealed before the end. So there should be four "K" before the end of the string. As the first card was picked from the "K" pile, when the play get to the last "K" in the string, there is no card in "K" pile and the play ends and does not continue. This contradicts with the assumption.
Therefore, if the string of the play ends with "K", then that will be a WIN game. Otherwise, It is "LOSE". And, the probability of winning the game is equal to the probability of having "K" at the end of the string which is equal to $\frac{1}{13}$.

Table 1 – Clock Solitaire Play rule

Clock solitaire

Draw a card from K pile (call it C)
Count=1
Repeat
  If possible
  Draw card from rank(C) pile
  Count=Count+1
  Continue
  else
  Break and say "Ooops! The pile is empty"
 	
If Count==5
  WIN
Else
  LOSE!

Stable Marriage
Let's continue to the Stable Marriage problem and provide the corresponding two facts for this problem.

1. Unpaired man choose next woman to propose by choosing women at random from those has not proposed.
2. Proposal stops once every women receives a proposal

Unfortunately, the first fact creates sequences of proposals with complicated distribution. For example, if the sequence of proposal is

Woman b, woman b, ?

Then the third one will not be “woman b” for sure and this disturbs the ideal uniform distribution which we hoped to have.
In order to have an algorithm that the probability of each woman proposed is uniform in each step, another algorithm has been suggested which is called “Amnesia proposal”.

Amnesia Proposal
In the Amnesia version of the algorithm, an unpaired man chooses next woman to propose uniformly from all women. Therefore we will have the following facts.

1. Amnesia version still produces stable matching and any repeated proposals are rejected.
2. Number of proposal made by Amnesia stochastically dominates number of proposal.

where stochastically dominates means

$$X \text{ stochastically dominates } Y \equiv P[X \geq x] \geq P[Y \geq x].$$

The Amnesia version of the algorithm provides an upper bound for the number of proposals resulted in stable marriage. From the previous lecture we know that this is a number between n and $n^2 - n + 1$, but we would like to know how much the average number of proposal is near to n or $n^2 - n + 1$.
In order to calculate the expected number of proposal, we first review Coupon Collector’s problem which is similar to the Amnesia version of the Marriage algorithm since in each step the probability of each event is equal. In Coupon Collector’s problem, there are n coupon and in each trial, a coupon is chosen at random. Let’s define Y to be the number of trail needed to collect at least one coupon at each type. The expected value of Y is

$$E[Y] = \sum_{y=0}^{\infty} P[Y = y]y$$

And let X_i be the number of trials performed when we have exactly i different coupon types, where p is the probability of having a new coupon i. Therefore

$$Y = X_0 + X_1 + \cdots + X_{n-1}$$

where $X_0 = 1$ and The expected value of X_i is estimated based on the following formula where

$$P[X_i = j] = (1-p)^{j-1}p$$
$$E[X_i] = \sum_j p[X_i = j]j = \sum (1-p)^{j-1}pj$$

This above formula is calculated using generating function. $E[X_i] = \frac{1}{p}$
In Amnesia algorithm $p = \frac{n-i}{n}$, therefore $E[X_i] = \frac{n}{n-i}$
From the linearity of the expectation, we have

$$E[A + B] = E[A] + E[B]$$

Therefore, we have

$$E[Y] = E[X_0] + E[X_1] + \cdots + E[X_{n-1}]$$
$$= 1 + \frac{n}{n-1} + \frac{n}{n-2} + \cdots + \frac{n}{1} = n\sum_{i=1}^{n}\frac{1}{i}$$

Where $\sum \frac{1}{i} = \ln(n)$, therefore we have

$$E[Y] = n\big(\ln(n) + O(1)\big) = nln(n) + O(n)$$
$$n \leq Y \leq nln(n)$$

However this does not tell too much about the distribution itself.
Based on Markov's Inequality for Z, a non-negative integer random variable, and $a > 0$

$$P[Z \geq a] \leq \frac{1}{a} E[Z]$$

Proof:

$$P[Z \geq a] = \sum_{j \geq a}^{\infty} P[Z = j] \leq \sum_{j=0}^{\infty} P[Z = j] \frac{j}{a} = \frac{1}{a} E[Z]$$

3 Conclusion

The expected running time over random inputs is an estimate of the running time complexity of the algorithm. There are other measures for characterizing the complexity of a Randomized input algorithm like variance which will be discussed in the following lectures.

References

[1] http://en.wikipedia.org/wiki/Stable_marriage_problem
[2] http://en.wikipedia.org/wiki/Clock_patience