# CPSC 500 Fundamentals of Algorithm Design and Analysis

Lecturer: Prof. Will Evans Date: September 30, 2013 Scriber: YuanFang Chi

## 1. Overview

The previous lecture has introduced the coupon collector's problem. This lecture explains the expectation and variance of this problem. Furthermore, this lecture also discusses the Chebyshev's Inequality, K-select problem, and selection via random sampling algorithm.

## 2. Coupon Collector's Problem

Let Y = the number of trials to collect all n coupons and $X_i$ = the number of trials performed when we have exactly i different coupon types. The previous lecture has shown that:

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

and

$$\mathrm{E}[Y] = n \ln n + \mathrm{O}(n)$$

But how close is Y to its expectations?

### 2.1. Variance

**Definition** The variance of random variable x is

$$Var[x] = \mathrm{E}[(x - \mathrm{E}[x])^2]$$

For independent x and y:

$$Var[x + y] = Var[x] + Var[y]$$

**Applying on coupon collector's problem**

In the coupon collector's problem, $X_0$, $X_0 \ldots X_{n-1}$ are mutually independent of each other. Therefore, the variance of Y can be given as:

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

Then, let p = the probability of getting a different coupon type, $p = \frac{n-i}{n}$. For general function $Var\ [X_i]$,

$$Var\ [X_i] = \frac{1-p}{p^2}$$

$$Var\ [X_i] = \frac{ni}{(n-i)^2}$$

$$Var[Y] = \sum_{i=0}^{n-1} Var\,[X_i]$$
$$= \sum_{i=0}^{n-1} \frac{ni}{(n-i)^2}$$
$$= n\left(n\sum_{i=1}^{n}\frac{1}{i^2} - H_n\right)$$
$$\approx n\left(n\frac{\pi^2}{6} - H_n\right) \leq 2n^2$$

So the variance is asymptotically $n^2$

## 3. Chebyshev's Inequality

**Definition** Chebyshev's inequality states that for the standard deviation k, less than $\frac{1}{k^2}$ of the distribution's values can be k away from the mean. Let $\sigma^2$ be the variance of X, Chebyshev's inequality is represented as

$$P[|X - \mathrm{E}[X]| \geq k] \leq \frac{\sigma^2}{k^2}$$

**Proof** Let Z = $(X - \mathrm{E}[X])^2$

$$E[Z] = Var[X] = \sigma^2$$

By Markov's Inequality,

$$P[Z \geq k^2] \leq \frac{1}{k^2}\,\sigma^2$$

$$P[Z \geq k^2] = P[|X - \mathrm{E}[X]| \geq k]$$

**Applying on coupon collector's problem**

For coupon collector's problem, $Var[Y] \approx n^2\frac{\pi^2}{6}$ and $E[Y] = nH_n$ are derived earlier, by Chebyshev's inequality

$$P[Y \geq bnH_n] \leq P[|Y - nH_n| \geq (b-1)nH_n]$$
$$\leq \frac{\frac{n^2\pi^2}{6}}{(b-1)^2 n^2 {H_n}^2}$$
$$\approx \frac{\pi^2}{6(b-1)^2(\ln n)^2}$$

As shown in the equation, the probability is dependent on n. Probability goes down when n gets larger. In 1961, Paul Erdos and Alfred Renyi proved that

$$\lim_{n\to\infty} P[Y \geq n\ln n + cn] = 1 - e^{-e^{-c}}$$

**Example** For c = 1, 2, 3, 4, the result of $1 - e^{-e^{-c}}$ is given below:

| | |
|---|---|
| 1 | 0.31 |
| 2 | 0.13 |
| 3 | 0.05 |
| 4 | 0.02 |

## 4. The K-Select Problem

**Definition** For input of n distinct real numbers $S$ $[1 \dots n]$, the K-Select problem describes how to find the $k^{th}$ smallest number in S. One obvious algorithm is to sort S in the increasing order and pick the $k^{th}$ element in S. However, this algorithm may take $\frac{n}{2}$ selections to find the median. A better algorithm of selection is to pick a pivot and partition S by it. Then, the algorithm determines which subset the $k^{th}$ element is in and then recurses on it. This algorithm is described in pseudo code below.

```
select(S, k){
        if |S| = 1 return S[1];
        p = GoodPivot(S);
        L = {S[i] | S[i]< p};
        R = {S[i] | S[i]> p};
        if |L| ≥ k return select(L, k);
        else if |L| = k-1 return p;
        else return select(R, k-|L|-1);
}
```

If the good pivot is picked randomly from S, the running time of the algorithm is expected to be: $< 4n$. Blum, Floyd, Pratt, Rivest and Tarjan introduced an algorithm to arrange S into groups of 5 and pick a median of each group. Then, select the median of those $\frac{n}{5}$ medians to be the good pivot. The running time of this algorithm is approximately $T(n) = T\left(\frac{n}{5}\right) + n + T\left(\frac{7n}{10}\right)$. We will show that there is a randomized algorithm that performs $1.5n + o\ (n)$ comparisons to find the median in expectation. There is a lower bound of $2n$ on all deterministic algorithms that find the median, which we won't show. Thus the randomized algorithm's expected performance is faster than any deterministic algorithm.

## 5. Selection via Random Sampling

An algorithm of selection via random sampling is introduced briefly.

Assume $k \in [n^{\frac{1}{4}}, n - n^{\frac{1}{4}}]$

1. Select $n^{\frac{3}{4}}$ numbers from S at random (with replacement), call the resulting sequence R.
2. Sort R O (n) time. (Reduce problem by factor $\frac{1}{n^{\frac{1}{4}}}$)
3. Let a be the $(\frac{k}{n^{\frac{1}{4}}} - \sqrt{n})$ smallest in R;
   Let b be the $(\frac{k}{n^{\frac{1}{4}}} + \sqrt{n})$ smallest in R.
4. Partition S into $S_L = \{x \in S | x < a\}, S_O = \{x \in S | a \leq x \leq b\}, S_R = \{x \in S | x > b\}$.
5. If $|S_L| \geq k$, then FAIL.
6. If $|S_L| + |S_O| < k$, then FAIL.
7. If $|S_O| > 4n^{\frac{3}{4}}$, then FAIL.
8. Sort $S_O$ and return $(k - |S_L|)$ smallest from $S_O$.