5. Probabilistic Analysis and Randomized Algorithms

Chun-Chao Yeh

Chapter 5: Probabilistic Analysis and Randomized Algorithm

From: Introduction to Algorithms,3ed,
by Cormen, Leiserson, Rivest & Stein

C.C. YEH

- Randomized vs. deterministic
- For a fixed input instance
  - Output is the same → deterministic
  - Output is vary from time to time → randomized
  - Note: this is only a rough description, not a formal definition

5.1 The Hiring problem

- Objective, e.g.
  - Find the best
  - Minimize number of times of hiring
  - …

```
HIRE-ASSISTANT(n)
best ← 0          ▷ candidate 0 is a least-qualified dummy candidate
for i ← 1 to n
    do interview candidate i
       if candidate i is better than candidate best
         then best ← i
              hire candidate i
```

Source: http://www.acetheinterview.com/

5.1 The Hiring problem

- Objective, e.g.
  - Find the best
  - Minimize number of times of hiring
  - …


```
HIRE-ASSISTANT(n)
1  best ← 0        ▷ candidate 0 is a least-qualified dummy candidate
2  for i ← 1 to n
3       do interview candidate i
4          if candidate i is better than candidate best
5             then best ← i
6                  hire candidate i
```

Cost: $O(nc_i + mc_h)$; n: number of candidate, m: number of changes (line 5)
C_i: interview cost;
C_h : hiring cost(layoff the current employee, and hire the new one ;

The parameter "m" is subject to change for different input instances.
So we focus on "m" only

Worse case v.s. average case analysis

- What is the number of "m"
  - ➔ that is how many times the if statement (line 4 in algorithm Hire-Assistant(n)) is true?
  - In worse case: N
    - E.g A=(1,2,3,4) for n=4
  - In average case: some what complicate
    - In general: =

$$\sum_{k=1}^{n} k * \Pr(X = k)$$

The prob. of k times hiring is required

average case analysis of the hiring-assistant algorithm

- How to get p(x=k)? Nontrivial problem, check ex. 5.2.1, 5.2.2 for some k!!
- An alternative method
  - Using recurrence relation
  - Assume
    - input $A=(a_1,a_2,\ldots,a_n)$ is a permutation of sequence [1:n]
    - Let f(n) is number of rehiring required for a input instance of size n.

$$f(n)=\begin{cases} f(n-1)+1, & \text{if } a_n = n \\ f(n-1), & \text{otherwise} \end{cases}$$

$$\begin{aligned}\bar{f}(n) &= (\bar{f}(n-1)+1) * p(a_n = n) + \bar{f}(n-1) * p(a_n \neq n) \\ &= (\bar{f}(n-1)+1)*(1/n) + \bar{f}(n-1)*((n-1)/n) \\ &= \bar{f}(n-1)+(1/n) = \bar{f}(n-2)+(1/(n-1))+(1/n) \\ &= \bar{f}(1)+1/2+1/3+\ldots+1/n = H(n) = \Theta(\ln n)\end{aligned}$$

5.2 indicator random variables

- Given a sample space S and an event A, Indicator random variable I{A}

$$\mathrm{I}\{\mathrm{A}\}=\begin{cases} 1, & \text{if A occurs} \\ 0, & \text{if A does not occurs} \end{cases}$$

Lemma 5.1:
Given a sample space S and an event A in the sample space S, let X_A=I{A}.
Then $E[X_A]$=Pr{A} (Pr{A}: probability of event A occurs)

$$\text{Proof}: \mathrm{E}[\mathrm{X}_{\mathrm{A}}] = 1\cdot \Pr\{A\} + 0 \cdot \Pr\{\overline{A}\} = \Pr\{A\}$$

Analysis of the hiring problem using indicator random variable

- Let
  - X: be the random variable whose value equals to the number of times we hire a new office assistant,
  - Xi: be the indicator random variable associated with the event in which the ith candidate is hired .
- Then

$$X = X_1 + X_2 + \dots + X_n, \{X_1, X_2, \dots, X_n\} \text{ are iid}$$

$$\mathrm{E[X]} = \sum_{x=0}^{n} x \Pr\{X = x\}$$

$$\mathrm{E[X]} = \mathrm{E}[\sum_{x=0}^{n} X_i] = \sum_{x=0}^{n} E[X_i] = \sum_{x=0}^{n} 1/i = \ln n + O(1)$$

$E[X_i] = 1/i$?

think about that if you have randomly select 4 poker card what is the probability the last one card is biggest?

ehow.com

```
HIRE-ASSISTANT(n)
1  best ← 0        ▷ candidate 0 is a least-qualified dummy candidate
2  for i ← 1 to n
3       do interview candidate i
4          if candidate i is better than candidate best
5             then best ← i
6                  hire candidate i
```

Exercises: 5.1-1, 5.1-2

- 5.1-1
  - Show that the assumption that we are always able to determine which candidate is best in line 4 of procedure HireAssistant implies that we know a total order on the ranks of the candidates.
    - O(n) for sorting ?? impossible
- 5.1-2
  - Describe an implementation of the procedure RANDOM(a,b) that only makes calls to RANDOM(0,1). What is the expected running time of your procedure, as a function of a and b.
    - r=a+(b-a)RANDOM(0,1)


```
HIRE-ASSISTANT(n)
1  best ← 0        ▷ candidate 0 is a least-qualified dummy candidate
2  for i ← 1 to n
3       do interview candidate i
4          if candidate i is better than candidate best
5             then best ← i
6                  hire candidate i
```

Exercises: 5.1-3

frontporchdenver.com

- Suppose that you want to output 0 with probability ½ and 1 with probability ½. At your disposal is a procedure Biased-Random, that outputs either 0 or 1. It outputs 1 with some probability p and 0 with probability 1-p, where 0<p<1, but you do not know what p is.
  - Give an algorithm that uses Biased-Random as a subroutine, and returns an unbiased answer, returning 0/1 with equal probability (1/2)
  - What is the expected running time of your algorithm as a function of p?
- ->code: 00,11->X, 01->0, 10->1

```
unBiasedRandom01(){
  While(1){
    x= BiasedRandom01() ;
    y= BiasedRandom01() ;
    if(x==y) continue
    return x
  }
}
```

$$p' = \Pr\{fail\} = \Pr\{xy = 00 \text{ or } 11\} = p^2 + q^2 \quad (q = 1 - p)$$

$$q' = \Pr\{succ\} = 1 - p' = 2p(1 - p) = 2pq$$

X = number of fail before first succ,

$\Rightarrow$ X is a geometric dist.

$$\Pr\{X = k\} = q'^k p'$$

$$E[X] = \sum_{k=0}^{\infty} k \Pr\{X = k\} = \sum_{k=0}^{\infty} k q'^k p'$$

$$= p' \sum_{k=0}^{\infty} k q'^k = p' \frac{1}{p'^2} = \frac{1}{p'}$$

Lemma 5.2 , exercise 5-2

- Lemma 5.2:
  - Assuming that the candidates are presented in a random order, algorithm HireAssistant has an average-case total hiring cost of $O(c_h \ln n)$

Exercises 5.2-1, 5.2-2, 5.2-3, 5.2-4, 5.2-5

Exercises 5.2-1, 5.2-2,

- Exercise 5.2-1: In HireAssistant, assuming that the candidates are presented in a random order, what is the probability that you will hire exactly one time? What is the probability that you will hire exact n times?
  - 1. the first one is the best → 1/n
  - 2. the input is well sorted -> 1/n!
- Exercise 5.2-2: In HireAssistant, assuming that the candidates are presented in a random order, what is the probability the you will hire exactly twice?
  - 1. if b[i] is best, then b[1] is second best in b[0:i]
    - Xi: indicator of the event: b[i] is best, and b[1] is second best in b[1:i], P(Xi=1)=P(Xi=1|b[i] is best)*P(b[i] is best)
    - =(1/i-1)*1/n, i>1,

Exercises 5.2-3

- Exercise 5.2-3:use indicator random variables to compute the expected value of the sum of n dice.
  - Xij: I{ dice i is with value of j}

Source: casinoreviewbank.com

$$X = \sum_{i=1}^{n} \sum_{j=1}^{6} jX_{i,j}$$

$$E[X] = E[\sum_{i=1}^{n} \sum_{j=1}^{6} jX_{i,j}] = \sum_{i=1}^{n} \sum_{j=1}^{6} jE[X_{i,j}]$$

$$\sum_{i=1}^{n} \sum_{j=1}^{6} j1/6 = 1/6 \sum_{i=1}^{n} \sum_{j=1}^{6} j = 1/6(21n) = \frac{7n}{2}$$

Exercises 5.2-4
hat-check problem

- Exercise 5.2-4: Use indicator random variables to solve the following problem, which is known as the hat-check problem. Each of n customers gives a hat to a hat-check person at a restaurant. The hat-check person gives the hats back to the customers in a random order. What is the expected number of customers that get back their own hat?
  - Xi: I{ customer i has his/her own hat back}

Source: about-australia-shop.com

$$X = \sum_{i=1}^{n} X_i$$

$$E[X] = E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathrm{E}[\mathrm{X}_i] = \sum_{i=1}^{n} 1/\mathrm{n} = 1$$

Exercises 5.2-5

- Exercise 5.2-5: Let A[1..n] be an array of n distinct numbers. If i<j and A[i]>A[j], then the pair (i,j) is called an ***inversion*** of A. Suppose that each element of A is chosen randomly, independently, and uniformly from the range 1 through n. Use indicator random variables to compute the expected number of inversions.
  - Xij: I{ the pair (i,j) is inversion}

$$X = \sum_{i>j} X_{ij}$$

$$E[X] = E[\sum_{i>j} X_{ij}] = \sum_{i>j} \mathrm{E}[\mathrm{X}_{ij}]$$

$$= \sum_{i>j} 1/2 = \frac{1}{2}\sum_{i>j} 1 = \frac{1}{2} x \frac{n(n-1)}{2} = \frac{n(n-1)}{4}$$

$method\ \ 2:$

$$f(n) = f(n-1) + \sum_{i=1}^{n} (i-1) P(A[n] \text{ is ith largest })$$

$$= f(n-1) + \sum_{i=1}^{n} (i-1)(1/n) = f(n-1) + \frac{1}{n} \cdot \frac{n(n-1)}{2}$$

$$\Rightarrow f(n) = f(n-1) + \frac{n-1}{2}$$

$$\Rightarrow f(n) = \frac{n(n-1)}{4}$$

5.3 randomized algorithms

```
RANDOMIZED-HIRE-ASSISTANT(n)
randomly permute the list of candidates
best ← 0        ▷ candidate 0 is a least-qualified dummy candidate
for i ← 1 to n
     do interview candidate i
        if candidate i is better than candidate best
           then best ← i
                hire candidate i
```

```
HIRE-ASSISTANT(n)
best ← 0        ▷ candidate 0 is a least-qualified dummy candidate
for i ← 1 to n
     do interview candidate i
        if candidate i is better than candidate best
           then best ← i
                hire candidate i
```

Deterministic version

Worst case and Average case costs:

Same as the deterministic version!!

Lemma 5.3

- Lemma 5.3: The expected hiring cost of the procedure RandomizedHireAssistant is $O(c_h \ln n)$

How to generate a random permuted sequence

- Required in line 1 of the "Randomized-Hire_Assistance(n)" algorithm

Source: londonist.com

```
PERMUTE-BY-SORTING(A)
    n ← length[A]
    for i ← 1 to n
          do P[i] = RANDOM(1, n^3)
    sort A, using P as sort keys
    return A
```

Lemma 5.4

The procedure produces a uniform random permutation of the input, assuming that all priorities are distinct.

Proof:

- hint
  - Show the probability of the occurrence of particular permutation generated by the procedure is 1/n!.
- Details:

$\mathrm{E}_i : event\ \{A[i]\ \text{receives the ith smallest priority}\}$

$\Pr\{E_i\} = 1/n\ (why\ ?)$

Note: assuming that all priorities are distinct!!.

→what is the prob not dist? Ex5.3.5

we want to know :

$$
\begin{aligned}
&\Pr\{\mathrm{E}_1 \cap \mathrm{E}_2 \cap \cdots \cap \mathrm{E}_n\} \\
&= \Pr\{\mathrm{E}_1\} \cdot \Pr\{\mathrm{E}_2 \cap \cdots \cap \mathrm{E}_n \mid \mathrm{E}_1\} \\
&= \Pr\{\mathrm{E}_1\} \cdot \Pr\{\mathrm{E}_2 \mid \mathrm{E}_1\} \cdot \Pr\{\mathrm{E}_3 \cap \cdots \cap \mathrm{E}_n \mid \mathrm{E}_1 \cap \mathrm{E}_2\} \\
&= \Pr\{\mathrm{E}_1\} \cdot \Pr\{\mathrm{E}_2 \mid \mathrm{E}_1\} \cdot \Pr\{\mathrm{E}_3 \mid \mathrm{E}_1 \cap \mathrm{E}_2\} \cdots \Pr\{\mathrm{E}_n \mid \mathrm{E}_{n-1} \cap \cdots \cap \mathrm{E}_1\} \\
&= \left(\frac{1}{n}\right)\left(\frac{1}{n-1}\right) \cdots \left(\frac{1}{2}\right)\left(\frac{1}{1}\right) = \frac{1}{n!}
\end{aligned}
$$

In-place version

```
RANDOMIZE-IN-PLACE(A)
1  n ← length[A]
2  for i ← 1 to n
3        do swap A[i] ↔ A[RANDOM(i, n)]
```

Proof: (for the in-place version)

- An alternative method
- Asuume:
  - input
    - $A=(a_1,a_2,\ldots,a_n)$: a permutation of sequence [1:n]
  - Let
    - $Pr(x_{i,j}=k)$ be the probability of A[j]=k after i-th iteration.
- Goal: prove that :
  - 1. A still a permutation of sequence [1:n].
  - 2. $Pr(x_{n,j}=k)=1/n$ for all j,k
- Proof:
  - 1. trivial
  - 2. see the followings.

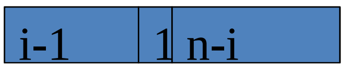
Proof: $Pr(x_{n,j}=k)=1/n$ for all j,k

- 1. properties:
  - After j-th iteration,
    - the value of A[j] is unchanged, that is $Pr(x_{n,j}=k)= Pr(x_{n-1,j}=k)= \ldots = Pr(x_{j,j}=k)$, for all j,k
- Proof: (informal)
  - Let
    - E_i be the event of $x_{n,i} \neq k$, and
    - M_i be the event of A[Random(i,n)]=k at i-th iteration, for some i and k.
  - We want to know
    - $Pr(x_{n,i}= k)= Pr (E_1 \cap E_2 \cap \ldots \cap E_{i-1} \cap M_i)$
    - $= Pr (M_i | E_1 \cap E_2 \cap \ldots \cap E_{i-1}) Pr (E_1 \cap E_2 \cap \ldots \cap E_{i-1})$
    - $= (1/(n-i+1))*((n-i+1)/n)=1/n$

proof

– 1. prove: $Pr (M_i | E_1 \cap E_2 \cap \ldots \cap E_{i-1}) = 1/(n-i+1)$

- Trivial:



- 2. $Pr (E_1 \cap E_2 \cap \ldots \cap E_{i-1})= (n-i+1)/n$
  - $P r(E_1 \cap E_2 \cap \ldots \cap E_{i-1})=Pr (E_2 \cap \ldots \cap E_{i-1} | E_1) Pr(E_1)$
  - $= Pr (E_3 \cap \ldots \cap E_{i-1} | E_1 \cap E_2) Pr(E_2 | E_1) Pr(E_1)$
  - $= Pr (E_{i-1} | E_1 \cap E_2 \cap \ldots \cap E_{i-2}) \ldots. Pr(E_2 | E_1) Pr(E_1)$
  - $=((n-i+1)/(n-i+2))* \ldots.*((n-2)/(n-1))*((n-1)/n)= (n-i+1)/n$

More formal proof can be done by induction

Exercises 5.3-5

- Exercise 5.3-5: Prove that in the array P in procedure PermuteBySorting, the probability that all elements are unique is at least 1-1/n?

$$\mathrm{P} = \frac{n^3 \cdot (n^3-1)\cdots(n^3-n+1)}{n^3 \cdot n^3 \cdots n^3} = \prod_{i=0}^{n-1} \frac{n^3-i}{n^3} \geq \prod_{i=0}^{n-1} \frac{n^3-n}{n^3} \geq \prod_{i=0}^{n-1} (1-\frac{1}{n^2})$$

$$= (1-\frac{1}{n^2})^n = 1 - C_1^n \frac{1}{n^2} + o(\frac{1}{n^2}) = 1 - \frac{1}{n} + o(\frac{1}{n^2}) \geq 1 - \frac{1}{n}$$

5.4 probabilistic analysis and further uses of indicator random variables

5.4.1: the birthday paradox

- Q: in a room of k persons, what is the probability that no two persons happens to have same birthday?
- A: B_k:the events of no two persons happens to have same birthday in a room of k persons.

$$A_i = event\{\text{person i's birthday is different from person j's for all } j < i\}$$

$$B_k = \bigcap_{i=1}^{k} A_i$$

$$\Pr\{B_k\} = \Pr\{B_{k-1}\}\Pr\{A_k \mid B_{k-1}\}$$

$$= \Pr\{B_{k-2}\}\Pr\{A_{k-1} \mid B_{k-2}\}\Pr\{A_k \mid B_{k-1}\}$$

$$\vdots$$

$$= \Pr\{B_1\}\Pr\{A_2 \mid B_1\}\ldots\Pr\{A_k \mid B_{k-1}\}$$

$$= 1\left(\frac{n-1}{n}\right)\left(\frac{n-2}{n}\right)\ldots\left(\frac{n-k+1}{n}\right) =$$

$$= \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{k-1}{n}\right)$$

$$\text{since } 1+x \le e^x$$

$$\Pr\{B_k\} \le e^{-1/n}e^{-2/n}\ldots e^{-(k-1)/n} = e^{-k(k-1)/2n}$$

Source: blogs.trb.com

Taylor series

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

$$e^x = \sum_{n=0}^{\infty} \frac{(e^x)^{(n)}|_{x=0}}{n!}(x-0)^n = \sum_{n=0}^{\infty} \frac{(e^x)|_{x=0}}{n!}(x)^n = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1} + \frac{x^2}{2!} + \ldots$$

the birthday paradox(2)

- Method 2, using indicator

$$X_{ij} = I\{\text{person i and person j have the same birthday}\}$$

$$E[X_{ij}] = 1/n$$

$$Let \quad X = \sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij}$$

$$E[X] = \sum_{i=1}^{k} \sum_{j=i+1}^{k} E[X_{ij}] = \binom{k}{2} 1/n = \frac{k(k-1)}{2n}$$

$$\Rightarrow when \quad k(k-1) > 2n, E[X] > 1$$

That is, if we have at least (2n)^1/2 +1 individuals in a room we can expect at least two of them to have the same birthday !!

5.4.2 balls and bins

- Consider the process of randomly tossing identical balls into b bins
  - How many balls fall in a given bin, if n balls are tossed?
    - n/b
  - How many balls must one toss, on the average, until a given bin contains a ball?
    - Pr(X=k)=p(1-p)^(k-1) (geometric distribution), p=1/b,

$$E[X] = \sum_{k=1}^{\infty} kp(1-p)^{k-1} = p\sum_{k=1}^{\infty} k(1-p)^{k-1} = p\frac{1}{p^2} = \frac{1}{p}$$

$$let \quad S(p) = \sum_{k=0}^{\infty}(1-p)^{k} = \frac{1}{1-(1-p)} = \frac{1}{p}$$

$$S'(p) = \frac{-1}{p^2} = \sum_{k=0}^{\infty} -k(1-p)^{k-1} = -\sum_{k=1}^{\infty} k(1-p)^{k-1}$$

$$\Rightarrow \sum_{k=1}^{\infty} k(1-p)^{k-1} = \frac{1}{p^2}$$

Source: repeatcrafterme.com

5.4.2 balls and bins (2)

coupon collector’s problem

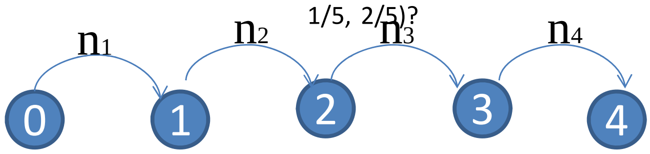
- 有一家飲料公司推出瓶蓋集字活動,每一瓶飲料的瓶蓋都有(恭, 喜, 發, 財)四字中其中一字, 集滿這四個字可以換獎金1000元.
  - 假設每一字出現的機率相同(皆為1/4), 請問平均要買幾瓶可己集滿這四個字??!!
  - variants:
    - How about for the cases of adding null word, for example, probability of occurrence of null word is ½, and the probability of occurrence of each of the four valid words is same as 1/8
    - How about for the cases of unequal probability, for exampl (1/5, 1/5, 1/5, 2/5)?

http://big5.made-in-china.com

5.4.2 balls and bins (2)

coupon collector’s problem

- 有一家飲料公司推出瓶蓋集字活動,每一瓶飲料的瓶蓋都有(恭, 喜, 發, 財)四字中其中一字, 集滿這四個字可以換獎金1000元.
  - 假設每一字出現的機率相同(皆為1/4), 請問平均要買幾瓶可已集滿這四個字??!!
  - variants:
    - » How about for the cases of adding null word, for example, probability of occurrence of null word is ½, and the probability of occurrence of each of the four valid words is same as 1/8?
    - » How about for the cases of unequal probability, for example (1/5, 1/5, 1/5, 2/5)?



$n = n_1 + n_2 + n_3 + n_4$

k state k: number of different k “words” you’ve collected so far

n_i Ni: a random value, how many tries you need to collect next word if now you have (i-1) different words in hands

http://big5.made-in-china.com

5.4.2 balls and bins (2)

coupon collector’s problem

– How many balls must one toss until every bin contains at least on ball? (coupon collector’s problem)

The hits can be used to partition the n tosses into stages.
The ith stage consists of the tosses after the (i-1)st hit until the ith hit.
Let n_i denote the number of tosses in the ith stage.
Thus the number of tosses required to get b hits is n=sum(n_i), 1≦i ≦b

$$E[n] = E[\sum_{i=1}^{b} n_i] = \sum_{i=1}^{b} E[n_i] = \sum_{i=1}^{b} b/(b-i+1) = bH_b = b(\ln b + O(1))$$

n_i = geometric distribution, with p = (b-i+1)/b,

x : geometric dist, with p, E[x]=1/p

5.4.3 streak

Source: mentalfloss.com

- Q: suppose you flip a fair coin n times.
  - What is the longest streak of consecutive heads that you expect to see?
    - That is, what is E[L], L is the length of longest streak of heads
    - ➔ An interesting problem!!
- Techniques:
  - It is hard to find the exact answer.
    - We use a bound to find out the approximate value.
  - Try to find a upper (lower) bound, then check if we can find the same lower (upper) bound.
  - Details:
    - somewhat complicate, please refer to the text book.
- Answer:

$\Theta(\lg n)$

proof :

$1. E[L] \le O(\lg n)$ (upper bound)

$2. E[L] \ge \Omega(\lg n)$ (low bound)

Streaks (upper bound)(1)

- Assume $A_{i,k}$ be the event that a streak of heads of length at least *k* begins with the *i*th coin flip.
- ➔

$$\Pr\{A_{i,k}\} = 1/2^k. \qquad \cdots(5.9)$$

$$For \quad k = 2\lceil \lg n \rceil,$$

$$\Pr\{A_{i,k}\} = 1/2^{2\lceil \lg n \rceil} \le 1/2^{2\lg n} = 1/n^2$$

$$\Pr\{\bigcup_{i=1}^{n-2\lceil \lg n \rceil+1} A_{i,2\lceil \lg n \rceil}\} \le \sum_{i=1}^{n-2\lceil \lg n \rceil+1} 1/n^2.$$

$$< \sum_{i=1}^{n} 1/n^2 = 1/n,$$

it imply that:

$\Pr\{\textit{length} \text{ of streak of heads} \ge k = 2\lceil \lg n \rceil\} < 1/n$

$\rightarrow$ *the* probability is small

$\rightarrow$ a hint to guess the upper bound is O(lgn)

Streaks (upper bound)(2)

- Goal: prove upper bound: O(lg n)
  - Assume :
    - L_j be the event that the longest streak of heads has length exactly j, for an instance.
    - And let L be the length of the longest streak.
- ➔

$$E[L] = \sum_{j=0}^{n} j \Pr\{L_j\} = \sum_{j=0}^{2\lceil \lg n \rceil - 1} j \Pr\{L_j\} + \sum_{j=2\lceil \lg n \rceil}^{n} j \Pr\{L_j\}.$$

$$< \sum_{j=0}^{2\lceil \lg n \rceil - 1} (2\lceil \lg n \rceil) \Pr\{L_j\} + \sum_{j=2\lceil \lg n \rceil}^{n} n \Pr\{L_j\},$$

$$= 2\lceil \lg n \rceil \sum_{j=0}^{2\lceil \lg n \rceil - 1} \Pr\{L_j\} + n \sum_{j=2\lceil \lg n \rceil}^{n} \Pr\{L_j\}$$

$$< 2\lceil \lg n \rceil * 1 + n * (1/n) = \mathrm{O}(\lg n)$$

Streaks (low bound)(1)

- Follows (5.9), we have $\Pr(A_{i,\lfloor (\lg n)/2 \rfloor}) = 1/2^{\lfloor (\lg n)/2 \rfloor} \geq 1/2^{(\lg n)/2} = n^{-1/2}$
  - ➔ not bad! So, we guest it is probably a good low bound.
- Tricks:
  - We partition the n coin flips into
    - at least $\lfloor n / \lfloor (\lg n)/2 \rfloor \rfloor$ groups
    - Each of the groups with $\lfloor (\lg n)/2 \rfloor$ consecutive flips,
  - and we bound the probability that no group comes up all heads.
  - By equation (5.9), the probability that the group starting in position *i* comes up all heads is $\Pr(A_{i,k}) >= n^{-1/2}$, k= $\lfloor (\lg n)/2 \rfloor$

$$\Pr\{A_{i,k}\} = 1/2^k$$

$$For \quad k = r\lceil \lg n \rceil,$$

$$\Pr\{A_{i,k}\} = 1/2^{r\lceil \lg n \rceil} \leq 1/2^{r \lg n} = 1/n^r$$

$$similarly, \quad r = 1/2 ==> \Pr\{A_{i,\lfloor (\lg n)/2 \rfloor}\} \geq 1/\sqrt{n}$$

Streaks (low bound)(2)

- The probability that a streak of heads of length at least floor((lg n)/2) doest not begin in position i is therefore at most 1- $n^{-1/2}$.
- Since the floor(n/((lg n)/2)) groups are formed from mutually exclusive, independent coin flips, the probability that every one of these groups fails to be a streak of length floor((lg n)/2) is at most O(1/n).

$$
\begin{aligned}
&(1-1/\sqrt{n})^{n/\lfloor(\lg n)/2\rfloor} \le (1-1/\sqrt{n})^{n/\lfloor(\lg n)/2\rfloor-1} \\
&\le (1-1/\sqrt{n})^{2n/\lg n-1} \\
&\le e^{-(2n/\lg n-1)/\sqrt{n}} \quad (\because 1+x \le e^x; here, x=-\frac{1}{\sqrt{n}}) \\
&= \mathrm{O}(e^{-\lg n}) = \mathrm{O}(1/n)
\end{aligned}
$$

Streaks (low bound)(3)

- Thus the probability that the longest streak exceeds floor((lg n)/2) is 1-O(1/n) at most.

$$\sum_{j=2\lfloor(\lg n)/2\rfloor+1}^{n} \Pr\{L_j\} \ge 1-\mathrm{O}(1/n)$$

- Finally, we have

$$
\begin{aligned}
&E[L] = \sum_{j=0}^{n} j\Pr\{L_j\} = \sum_{j=0}^{\lfloor(\lg n)/2\rfloor} j\Pr\{L_j\} + \sum_{j=\lfloor(\lg n)/2\rfloor+1}^{n} j\Pr\{L_j\}. \\
&\ge \sum_{j=0}^{\lfloor(\lg n)/2\rfloor} 0*\Pr\{L_j\} + \sum_{j=\lfloor(\lg n)/2\rfloor+1}^{n} \lfloor(\lg n)/2\rfloor * \Pr\{L_j\}, \\
&= o + \lfloor(\lg n)/2\rfloor \sum_{j=\lfloor(\lg n)/2\rfloor+1}^{n} \Pr\{L_j\} \ge \lfloor(\lg n)/2\rfloor(1-\mathrm{O}(1/n)) \\
&= \Omega(\lg n)
\end{aligned}
$$

streaks

- How do you know $\Theta(\lg n)$ it is a good guess?!
- Let X_{ik}=I{A_{ik}},
  - the indicator random variable associated with a streak of heads of length at least k beginning with the ith coin flip.

$$\mathrm{E}[X] = E[\sum_{i=1}^{n-k+1} X_{ik}] = \sum_{i=1}^{n-k+1} E[X_{ik}] = \sum_{i=1}^{n-k+1} \Pr(A_{ik})$$

$$= \sum_{i=1}^{n-k+1} \frac{1}{2^k} = \frac{n-k+1}{2^k}$$

$$k = 1, \frac{n-k+1}{2^k} = \frac{n}{2}$$

$$k = n, \frac{n-k+1}{2^k} = \frac{1}{2^n}$$

$$k = c\lg n, \frac{n-k+1}{2^k} = \frac{n - c\lg n + 1}{n^c} = \frac{1}{n^{c-1}} - \frac{c\lg n + 1}{n^c} = \Theta(1/n^{c-1})$$

5.4.4. the on-line hiring problem

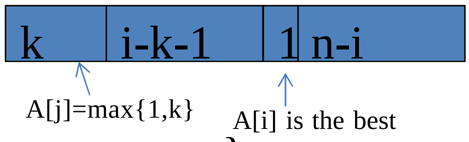
- What is the probability to hire the best candidate, given k,n?

```
ON-LINE-MAXIMUM(k, n)
bestscore ← −∞
for i ← 1 to k
      do if score(i) > bestscore
            then bestscore ← score(i)
for i ← k + 1 to n
      do if score(i) > bestscore
            then return i
return n
```

Source: employeescreeningblog.com

on-line hiring problem

- Let S_i be the event we successfully interview the best candidate at i-th candidate.
  - What is $Pr(S_i)$?
- $Pr(S_i)=0$ if $i<=k$ ➔ trivial
- Again assume
  - input $A=(a_1,a_2,\ldots,a_n)$ is a permutation of sequence [1:n]
  - threshold=max(A[1:k])



- We want:
  - for each j , k<j<i, A[j]<threshold, and threshold≠n
    - $Pr(S_i)$=Pr (a[i]=n and b=max(A[1:i-1]) is in A[1:k])
    - = Pr(a[i]=n)*Pr (b=max(A[1:i-1]) is in A[1:k])
    - = 1/n * k/(i-1)=k/(n*(i-1));
  - So, Pr(S)=sum(i=1,n) $Pr(S_i)$=sum(i=k+1,n) $Pr(S_i)$
    - =k/n(H(n-1)-H(k-1)) ~k/n(ln n – ln k)

on-line hiring problem

- Finding a setting of k to maximize the Pr(S), can be done by differential the expression k/n(ln n- ln k) w.r.t k,
  - Then we have k=n/e, and the expression turns to be 1/e

$$\frac{d(\frac{k}{n}(\ln n-\ln k))}{dk}=0$$

$$\frac{1}{n}(\ln n-\ln k)+\frac{k}{n}(-\frac{1}{k})=0$$

$$\ln n-\ln k=1, k=e^{\ln n-1}=\frac{n}{e}$$

Exercises 5.4-2, 5.4-6

- Exercise 5.4-2: suppose that balls are tossed into b bins. Each toss is independent, and each ball is equally likely to end up in any bin. What is the expected number of ball tosses before at least on of the bins contains two balls?

X_i : *event* {i -1 balls are tossed and non of them are in the same bins}
and {the last ball is tossed into one of the bins tossed before}

$$X = \sum_{i=2}^{b+1} \mathrm{X}_i$$

$$\Pr\{\mathrm{X}_i = 1\} = \left(\frac{P_{i-1}^{b}}{b^{i-1}}\right)\left(\frac{i-1}{b}\right) = \frac{P_{i-1}^{b}}{b^{i}}(i-1)$$

$$E[X] = E[\sum_{i=2}^{b+1} \mathrm{X}_i] = \sum_{i=2}^{b+1} \mathrm{E}[\mathrm{X}_i] = \sum_{i=2}^{b+1} \mathrm{i}\Pr\{\mathrm{X}_i = 1\}$$

$$= \sum_{i=2}^{b+1} \mathrm{i}(\mathrm{i}-1)\left(\frac{P_{i-1}^{b}}{b^{i}}\right)$$

Source: repeatcrafterme.com

Exercises 5.4-2, 5.4-6

- Exercise 5.4-6: suppose that n balls are tossed into n bins, where each toss is independent and the ball is equally likely to end up in any bin. What is the expected number of empty bins? What is the expected number of bins with exactly one ball?

X_i : I {bin i is empty after n toss}

$$X = \sum_{i=2}^{b+1} \mathrm{X}_i$$

$$\Pr\{\mathrm{X}_i = 1\} = \left(\frac{n-1}{n}\right)^n, E[\mathrm{X}_i] = \left(\frac{n-1}{n}\right)^n$$

$$E[X] = E[\sum_{i=1}^{n} \mathrm{X}_i] = \sum_{i=1}^{n} \mathrm{E}[\mathrm{X}_i] = \sum_{i=1}^{n} \left(\frac{n-1}{n}\right)^n = n\left(\frac{n-1}{n}\right)^n$$

X_i : I {bin i has exactly one ball after n toss}

$$X = \sum_{i=2}^{b+1} \mathrm{X}_i$$

$$\Pr\{\mathrm{X}_i = 1\} = \binom{n}{1}\frac{1}{n}\left(\frac{n-1}{n}\right)^{n-1} = \left(\frac{n-1}{n}\right)^{n-1}, E[\mathrm{X}_i] = \left(\frac{n-1}{n}\right)^{n-1}$$

$$E[X] = E[\sum_{i=1}^{n} \mathrm{X}_i] = \sum_{i=1}^{n} \mathrm{E}[\mathrm{X}_i] = \sum_{i=1}^{n} \left(\frac{n-1}{n}\right)^{n-1} = n\left(\frac{n-1}{n}\right)^{n-1}$$

Problem 5.2.f

- Problem 5.2.f: suppose that are k>0 indices i such that A[i]=x. what is the expected running time of a linear search over array A[1..n] for x.
  - Ans: E[X]=(n+1)/(k+1)

K=1, trivial,
E[X]=(n+1)/2, k>1?

Source: yaymicro.com

$$\text{X} = \text{i : event \{number of i searchs before first x encounted\}}$$

$$\Pr\{\text{X} = i\} = \left(\frac{\binom{n-i}{k-1}}{\binom{n}{k}}\right), 1 \le i \le n-k+1$$

$$E[X] = \sum_{i=1}^{n-k+1} \text{i}\Pr\{\text{X} = i\} = \sum_{i=1}^{n-k+1} \text{i}\left(\frac{\binom{n-i}{k-1}}{\binom{n}{k}}\right) = \frac{n+1}{k+1}$$

$$\text{or, } \sum_{i=1}^{n-k+1} \text{i}\binom{n-i}{k-1} = \frac{n+1}{k+1}\binom{n}{k} = \binom{n+1}{k+1} \text{(proof)}$$

proof

- We want to prove:

$$\sum_{i=1}^{n-k+1} \text{i}\binom{n-i}{k-1} = \frac{n+1}{k+1}\binom{n}{k} = \binom{n+1}{k+1}$$

$$\text{Pascal's identity:}$$

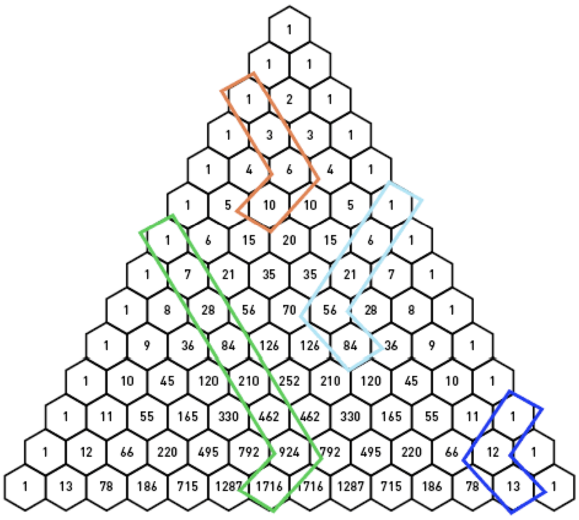
$$c(n+1,k) = c(n,k-1) + c(n,k), \ \text{ where c(i, j)} = \binom{\text{i}}{\text{j}}$$

$$\textit{extention:}$$

$$\sum_{k=0}^{r} c(n+k,k) = \sum_{k=0}^{r} c(n+k,n) = c(n+r=1,r)$$

Pascal's ide

HOCKEY STICKS IN PASCAL'S TRIANGLE



Source: learner.org

Pascal's identity :

$$c(n+1,k) = c(n,k-1) + c(n,k), \qquad \cdots(1)$$

where $c(i, j) = \binom{i}{j}$

extention :

$$\sum_{k=0}^{r} c(n+k,k) = \sum_{k=0}^{r} c(n+k,n) = c(n+r=1,r)$$

prove by Pascal's identity (1) :

$$\sum_{k=0}^{r} c(n+k,k) = c(n,0) + c(n+1,1) + c(n+2,2) + \ldots + c(n+r,r)$$
$$= [c(n+1,0) + c(n+1,1)] + c(n+2,2) + c(n+3,3) + \ldots + c(n+r,r)$$
$$= [c(n+2,1) + c(n+2,2)] + \ldots + c(n+r,r)$$
$$= [c(n+3,2) + c(n+3,3)] + \ldots + c(n+r,r)$$
$$= c(n+3,3) + \ldots + c(n+r,r) = \ldots$$
$$= c(n+r,r-1) + c(n+r,r) = c(n+r+1,r) \qquad \cdots(2)$$

$$\sum_{k=0}^{r} kc(n+k,k) = rc(n+r+1,r) - c(n+r+1,r-1)$$

$$\sum_{k=0}^{r} kc(n+k,k) = rc(n+r+1,r) - c(n+r+1,r-1)$$

proof : from (1) and (2)

$$\sum_{k=0}^{r} kc(n+k,k) =$$
$$= \sum_{k=1}^{r} c(n+k,k) + \sum_{k=2}^{r} c(n+k,k) + \sum_{k=3}^{r} c(n+k,k) + \ldots + \sum_{k=r}^{r} c(n+k,k)$$
$$= (c(n+r+1,r) - \sum_{k=0}^{0} c(n+k,k)) + (c(n+r+1,r) - \sum_{k=0}^{1} c(n+k,k))$$
$$+ (c(n+r+1,r) - \sum_{k=0}^{2} c(n+k,k)) + \ldots + (c(n+r+1,r) - \sum_{k=0}^{r-1} c(n+k,k))$$
$$= rc(n+r+1,r) - (c(n+1,0) + c(n+2,1) + c(n+3,2) \ldots + c(n+r,r-1))$$
$$= rc(n+r+1,r) - \sum_{k=0}^{r-1} c(n+1+k,k) = rc(n+r+1,r) + c(n+1+r-1+1,r-1)$$
$$= rc(n+r+1,r) - c(n+r+1,r-1) \qquad \cdots(3)$$

$$proof: \sum_{i=1}^{n-k+1} \mathrm{i}\binom{n-i}{k-1} = \frac{n+1}{k+1}\binom{n}{k} = \binom{n+1}{k+1}$$

$$\sum_{i=1}^{n-k+1} \mathrm{i}c(\mathrm{n-i,k-1}) = \sum_{i=0}^{n-k} (\mathrm{n-k+1-i})c(\mathrm{k-1+i,k-1}) = \sum_{i=0}^{n-k} (\mathrm{n-k+1-i})c(\mathrm{k-1+i,i})$$

$$= (\mathrm{n-k+1})\sum_{i=0}^{n-k} c(\mathrm{k-1+i,i}) - \sum_{i=0}^{n-k} \mathrm{i}c(\mathrm{k-1+i,i})$$

$$= (n-k+1)c(n,k) - ((n-k)c(n,k) - c(n,k+1))$$

$$= c(n,k) + c(n,k+1) = c(n+1,k+1)$$

from (2)

$$\sum_{i=0}^{n-k} c(\mathrm{k-1+i,i})$$

$$= c(k-1+n-k+1, n-k)$$

$$= c(n,n-k) = c(n,k)$$

from (3)

$$\sum_{i=0}^{n-k} ic(\mathrm{k-1+i,i})$$

$$= (n-k)c(k-1+n-k+1, n-k)$$

$$-c(k-1+n-k+1, n-k-1)$$

$$= (n-k)c(n,n-k) - c(n,n-k-1)$$

$$= (n-k)c(n,k) - c(n,k+1)$$