Randomized Algorithms

1. Preliminaries

1.1. Why randomization?

During the course we will see algorithms which:

→ exhibit smaller runtimes / smaller space consumption than det. strategies

→ are easier to implement / more elegant

→ good strategies in games and unknown environments

1.2. Basic Stochastics

X - random variable which can take discrete numerical values

$E(X)$ - expected value of X

Def: Let $x_1, \dots, x_k$ be the finite set of values X can take, and $p_1, \dots, p_k \in [0,1]$ the respective probabilities of realization $P(X = x_i)$.

Then the expected value of X is

$$E(X) = \sum_{i=1}^{a} x_i \cdot p_i$$

Example (Dice): $x_1 = 1,\ x_2 = 2, \dots, x_6 = 6$

$$p_1 = p_2 = \dots = p_6 = \frac{1}{6}$$

$$E(X) = \frac{1}{6} \sum_{i=1}^{6} i = \frac{21}{6} = 3.5$$

First Success

X - random variable, x_i - success

p - success probability

$q = P(X \neq x_i) \Rightarrow q = 1-p$

Y - random variable counting the steps until first success

$$j \in \mathbb{N}$$

$$P(Y=j) = q^{j-1} \cdot p$$

$$E(Y) = \sum_{j=1}^{\infty} j \cdot P(Y=j) = \sum_{j=1}^{\infty} j \cdot (1-p)^{j-1} \cdot p$$

$$= \frac{p}{1-p} \sum_{j=1}^{\infty} j \cdot (1-p)^{j} = \frac{\cancel{p}^{\,1}}{\cancel{1-p}_{\,1}} \cdot \frac{\cancel{1-p}^{\,1}}{\cancel{p^2}_{\,p}} = \frac{1}{p}$$

Lemma: If a random exp. with success prob. $p > 0$ is performed repeatedly with the outcomes being independent of each other, the exp. number of rounds for first success is $\frac{1}{p}$.

Example (First 6): $p = \frac{1}{6}$ $q = \frac{5}{6}$

$$\Rightarrow E(Y) = 6$$

Linearity of Expected Value

multiple random variables $X_1, X_2, \ldots, X_a$
then
$$E(X_1 + X_2 + \ldots + X_a) = E(X_1) + E(X_2) + \ldots + E(X_a)$$

Exercise: Prove the linearity of the expected value.

Example (Guessing cards) You have n distinct cards. You want to predict them one-by-one.

Case A) Completely drunk (memory-less)

X - number of correctly guessed cards

$E(X)$?

$X_1 = \begin{cases} 1 & \text{if you guessed card 1 right} \\ 0 & \text{otherwise} \end{cases}$

X_2

$\vdots$

X_n

$$E(X) = E(X_1 + X_2 + \dots + X_n) = \sum_{i=1}^{n} E(X_i)$$

$$P(X_i = 1) = \frac{1}{n} \quad E(X_i) = 1 \cdot \frac{1}{n} + 0 \cdot \frac{n-1}{n} = \frac{1}{n}$$

$$\Rightarrow E(X) = n \frac{1}{n} = 1$$

Case B) not drunk (with memory)

$\Rightarrow$ now the pool of cards to choose from shrinks in every round

$$E(X_i) = P(X_i = 1) = \frac{1}{n-i+1}$$

$$E(X) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \frac{1}{n-i+1} = \sum_{i=1}^{n} \frac{1}{i} = H(n)$$

$$\ln(n+1) < H(n) < 1 + \ln(n)$$

$$H(n) \in \Theta(\log n)$$

Example (Collecting Goodies) Every time you shop at a certain shop you get 1 of n toys for free (you cannot choose which). How often do we have to shop there expectedly to collect all n toys?

X – number of visits till all toys are collected

$E(X)$?

$X_1, X_2, X_3, \ldots$

X_i – counts the number of steps to go from $i-1$ to i toys

- chance of success in phase i:

$$p = \frac{n-i+1}{n}$$

$$\Rightarrow E(X_i) = \frac{n}{n-i+1}$$

$$\Rightarrow E(X) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \cdot \sum_{i=1}^{n} \frac{1}{n-i+1}$$

$$= n \cdot H(n) \in O(n \log n)$$

Conditional Probability

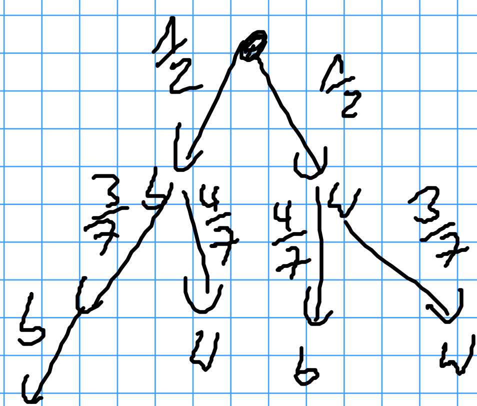
Urn Model

Coloured balls (e.g. 4 black ones, 4 white ones). If you pick a ball, the colour is only revealed after removal from the urn. Then the ball is not returned. What are the chances to pick a white ball in round 1, 2, 3, ...?

$$P(w) = \frac{1}{2}$$

$$P(w \text{ in round } 2 \mid w \text{ in round } 1) = \frac{3}{7}$$

$$P(w \text{ in round } 2 \mid b \text{ in round } 1) = \frac{4}{7}$$



$$P(w \text{ in round } 5 \mid w \text{ in round } 1,2,3,4) = 0$$

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (P(B) > 0)$$

$$E(X|Y) = \sum_{i=1}^{n} x_i \; P(X = x_i \mid Y)$$

1.3. Las Vegas and Monte Carlo Algorithms

Example (Drug Detection) n lockers and $\frac{n}{2}$ of the pupils hide drugs in them

Break open lockers until you find some drugs.

How many lockers are required?

Best case: 1 Worst case: $\frac{n}{2}+1$

Deterministic Strategy: open lockers in order of their numbers

Randomized Strategy 1: permute locker numbers, attach them in that order

Randomized Strategy 2: choose k lockers and open them

↳ chance to find drugs?

p – Propability of drugs in 1 locker is $\frac{1}{2}$

p(drugs are found) $= 1 - P$(no drugs were found)

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

Rand. Strategy 1 is an example of a Las Vegas algorithm:
- output is always correct
- runtime depends on randomization

Rand. Strategy 2 is an example of a Monte Carlo algorithm:
- runtime does not depend on randomization
- Chance of failure

Example (Find Large Number) Given an array of n elements, we search for any element $\geq$ median.

- Det. Strategy: * sort $O(n \log n)$
  pick sth. in second half $O(1)$

  $O(n \log n)$

  * single scan always remembering the maximum $O(n)$

- Randomized:

Monte Carlo Find

- pick k elements u.a.r. (uniformly at random)
- find the max of these elements
- return max

→ runtime: $O(k)$

Probability that a random element is not equal to or larger than the median is $\leq \frac{1}{2}$. The prob. of all k elements being incorrect is $\leq \frac{1}{2}^k$. So if we set $k = c \cdot \log n$ for some positive constant c, the chance of error is n^{-c}. If $k=20$ the chance is smaller than 1:1 Mio. and for $k=25$ 1: 15 Mio. ⇒ Prob. solely depends on k, so the error rates are true for e.g. a quadrillion elements ⇝ cost of deterministic strategy would be $\frac{1}{2}$ quadrillion.

Example (Find Repeated Element) Given array of size $n > 4$, filled with $\frac{n}{2}$ distinct elements, $\frac{n}{2}$ times a repeated element
(1, 2, 3, 4, 4, 4)

Find that repeated element.

Det. Strategy: Sort $O(n \log n)$ $\rightsquigarrow$ scan to find equal neighbours $O(n)$

$$\underbrace{\qquad\qquad\qquad\qquad}_{O(n \log n)}$$

Rand. Strategy

Las Vegas Find Repeated Element

<u>while</u> true <u>do</u>

pick i u.a.r. in $\{1, \dots, n\}$

pick j u.a.r. in $\{1 \dots, n\} - \{i\}$

<u>if</u> $(A[i] == A[j])$ <u>then</u>

<u>return</u> $A[i]$

Correctness is obvious. What is the expected runtime?

$$P(\text{success}) = \frac{n/2}{n} \cdot \frac{\frac{n}{2} - 1}{n-1} \geq \frac{1}{4}$$

So according to our Lemma we need to wait expectedly 4 rounds for success.

1.3.2 From LV to MC and Back

Any LV-algo. can be turned into a MC-alg.

Given a LV-alg. with expected runtime $T(n)$. To get a fixed runtime, we just stop after $\alpha \cdot T(n)$ time for some $\alpha \in \mathbb{R}^+$. If the alg. returned an answer, it is correct for sure. Otherwise we return 'error'.

What is the probability of error?

Def (Markov Inequality) For non-neg. random variable X we have

$$P(X > t) \leq \frac{E(X)}{t}$$

Exercise: Proove it.

We reformulate MI slightly

$$P(X > t \cdot E(X)) \leq \frac{1}{t}$$

Now if X is the runtime of our LV alg. and $E(X)$ the expected value, here $T(n)$ we get

$$P(X > \alpha \cdot T(n)) \leq \frac{1}{\alpha}$$

$\Rightarrow$ probability to fail