Cheat Sheet 2

Maha Alkhairy

① probability + expectations ② randomized algorithms

③ Network flow ④ Linear programing

① probability + expectations

$|A \cup B| = |A| + |B|$ — A and B are sets

$n!$ permutations of a set of n elements

$\binom{n}{k}$ choose k elements from n items

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$$

$Pr[A \cup B] = Pr[A] + Pr[B] - P[A \cap B]$ ← if mutually exclusive

$Pr[A \cap B] = P[A] \cdot P[B|A] = P[B] \cdot P[A|B]$

$= P[A] \cdot P[B]$ ← if independent

if takes on natural numbers

$$E[X] = \sum_x x \cdot Pr[X = x] = \sum_{i=0}^{\infty} i \cdot Pr[X = i]$$

$E[X+Y] = E[X] + E[Y]$ ← linearity of expectation

$E[XY] = E[X]E[Y]$ ← X, Y are independent

$Var[X] \geq 0$ $\quad Var[X] = E[X^2] - E[X]^2$

$Var[aX] = a^2 Var[X]$

$$Var\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} Var[X_i]$$

bb) $m > 6n \ln(n)$: $k = \frac{m}{n} + \sqrt{6 \ln(n) \left(\frac{m}{n}\right)}$

Z_i^j = expected # ball in bin j $\quad Z_i^j = 1$ ← if ball i in bin j, 0 otherwise

$$E[Z^j] = \sum_{i=1}^{m} 1 \cdot Pr[Z_i^j = 1] = \frac{m}{n}$$

union bound:-

$Pr[Z^{max} > k] = Pr[Z^1 \geq k]$ or $Pr[Z^2 \geq k]$ or ... $Pr[Z^n > k]$

fullest bin has at most k balls

$\leq n \, Pr[Z^j \geq k]$ ← union bound

use Chernoff bound

$Pr[Z^{max} \leq k] = 1 - Pr[Z^{max} > k]$

Chernoff bound $Pr[Z - \mu \geq \epsilon\mu] \leq e^{-\epsilon^2 \mu/3} \quad \forall \, 0 \leq \epsilon \leq 1$

$k = \mu + \epsilon\mu = \frac{m}{n} + \sqrt{6 \ln(n) \left(\frac{m}{n}\right)} \geq 6 \ln(n) + \sqrt{6 \ln^2(n) \cdot 6 \ln(n)}$

ϵ ... get ϵ and solve Chernoff bound from plug into union bound

② randomized Algorithms

average runtime may be better than worse case runtime.

$Pr[\text{"X at least k times"}] = 1 - Pr[\text{not X k times}]$

$T(n)$ is random, want $E[T(n)]$

$$E[T(n)] = \sum_{k=1}^{\infty} k \, Pr[T(n) = k]$$

Quicksort :- → $\Theta(n^2)$ worst case

- Choose pivot $\in p$ randomly
- partition to two subarrays $e_i \geq p$; $e_j < p$
- sort two subarrays recursively

runtime: $O(1) \times$ number of comparisons

$X_{ij} \in 1$ if i^{th} smallest elem compared j^{th} smallest elem, 0 otherwise

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} \leftarrow E[X] = \text{\# of comparisons}$$

$n \log n$ ← expected runtime of randomized quick sort

Balls and Bins

m balls n bins

$$Pr[B_1 = k] = \binom{n}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k}$$

B_1 : # balls in bin 1

$$Pr[B_1 \geq k] = 1 - Pr[B_1 \leq k] = \left(\frac{n}{k}\right)\left(\frac{1}{n}\right)^k \leq \frac{1}{k!}$$

$Pr[\exists \text{ bin at least } k \text{ balls}]$:

$$\sum_{i=1}^{n} Pr[B_i \geq k] \leq n \, Pr[B_1 \geq k] \leq n/k!$$

union bound

$m = n$: upper bound each ball in diff bin $= 1/e$

$Pr[\text{a bin gets} \geq 2 \text{ balls}] = 1 - 1/e$

③ Network flow :-

f is the flow, c is the capacity

s is the source, t is the sink

$\forall \, v \neq s, t$

$$\sum_u f(u \to v) = \sum_w f(v \to w) \leftarrow \text{conservation constraint}$$

$|f|$ = net flow into t

$$|f| = \sum_u f(u \to t) - \sum_w f(t \to w)$$

augmenting path p is a simple path from s to t in the residual graph G_f

$0 \leq f(e) \leq c(e) \quad \forall e \in E$

runtime: $O(EVC)$ (E = # edges, V = # nodes, C = max capacity)

Residual Graph($G(V,E)$, f):

$E_f \leftarrow \emptyset$

for each edge $(u \to v) \in E$:-

  if $f(u \to v) < c(u \to v)$:

    $E_f \leftarrow E_f \cup \{(u \to v)\}$

    $c_f(u \to v) \leftarrow c(u \to v) - f(u \to v)$

  if $f(u \to v) > 0$:-

    $E_f \leftarrow E_f \cup \{(v \to u)\}$

    $c_f \leftarrow f(u \to v)$

return $(G_f(V, E_f), c_f(\cdot))$

Theorem: There is no path from s to t in residual graph (G_f for some feasible flow x in some network)

then:

① flow x is maximum flow

② let S be the set of all nodes reachable from s in residual graph (including s itself). Let T be the set of all other nodes. then (S,T) is a minimum cut and has capacity equal to the value of x

randomized algs

- One where certain decisions are made based on outcomes of coin flips made in algorithm.
- Analysis is done with inputs assuming anything about input distribution
- Done in spite of all possible outcomes for coin flips made in algorithm

Las Vegas: always correct, run time is random variable

Monte Carlo: decision probably correct with very high probability

Maxflow mincut theorem

$G=(V,E)$ f is flow s,t is source and sink :- Following conditions are equivalent :-

1) f is max flow in G
2) G_f contains no augmenting path
3) $|f| = C(S,T)$ for some cut of G

by Choosing augmenting path using BFS
runtime Maxflow = $O(E^2V)$

$|f| = C(S,T)$ iff
* f saturates every edge from S to T
* f avoids every edge from T to S
→ maximal

$|f| \leq C(S,T)$ general

(LP) linear Programing

Primal LP - canonical form:

$\max\ c^T x$
$s.t\ Ax \leq b$
$x \geq 0$

n constraints
d dimentions / var
$x \in \mathbb{R}^d$; $A \in \mathbb{R}^{n \times d}$
$b \in \mathbb{R}^n$; $c \in \mathbb{R}^d$

Dual LP:

$\min\ b^T y$
$s.t\ A^T y \geq c$
$y \geq 0$
$y \in \mathbb{R}^n$

Duality theorem
$c^T x \leq b^T y$
x^* optimal y^* optimal
$c^T x^* = b^T y^*$

general LP

$\max \sum_{j=1}^{d} c_j x_j$

s.t

$\sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad \forall i = 1,2,\dots p$

$\sum_{j=1}^{d} a_{ij} x_j = b_i \quad \forall i = p+1, p+2, \dots q$

$\sum_{j=1}^{d} a_{ij} x_j \geq b_i \quad \forall i = q+1, q+2, \dots n$

General form to Canonical

1. $\sum_{j=1}^{d} a_{ij} x_j = b_i$ → $\sum_{j=1}^{d} a_{ij} x_j \leq b_i$, $\sum_{j=1}^{d} a_{ij} x_j \geq b_i$

2. replace $\sum_{j=1}^{d} a_{ij} x_j \geq b_i$ by $\sum_{j=1}^{d} (-a_{ij}) x_j \leq -b_i$

3. replace x_j by $x_j^+ - x_j^-$ and add $x_j^+ \geq 0$; $x_j^- \geq 0$

Primal	Dual
$\max c^T y$	$\min b^T y$
$\sum_j a_{ij} x_j \leq b_i$	$y_i \geq 0$
$\sum_j a_{ij} x_j \geq b_i$	$y_i \leq 0$
$\sum_j a_{ij} x_j = b_i$	N/A
$x_j \geq 0$	$\sum_i a_{ij} y_i \geq c_j$
$x_j \leq 0$	$\sum_i a_{ij} y_i \leq c_j$
N/A	$\sum_i a_{ij} y_i = c_j$
$x_j = 0$	N/A

feasible point:- $x \in \mathbb{R}^d$ satisfies all constraints
feasible region:- Set of all feasible points
Vertex: a point in region that makes d inequalities tight
Vertex is feasible: if it is a vertex and a feasible point
n-neighbors:- two vertices are neighbors if they share $d-1$ equations

Example LP and dual — max matching

$e \in E$ $x_e \in 0,1$ denote whether edge e is in maximal matching

primal:
$\max \sum_{e \in E} x_e$
s.t.
$\sum_{(u,v) \in E} x_{(u,v)} \leq 1 \quad \forall_v \in A \cup B$
$x_e \geq 0 \quad \forall e \in E$

dual:
$\min \sum_{v \in A \cup B} x_v$
$x_u + x_v \geq 1 \quad \forall (u,v) \in E$
$x_v \geq 0 \quad \forall v \in V$

edges
vertex cover