Scilab Textbook Companion for Discrete Mathematics by S. Lipschutz, M. Lipson And V. H. Patil[1]

Created by
Neerja Bahuguna
B.Tech. (pursuing)
Computer Engineering
Echelon Institute of Technology, Faridabad
College Teacher
Prashant Gupta, E. I. T. ,Faridabad
Cross-Checked by
Prashant Dave, IIT Bombay

August 10, 2013

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Discrete Mathematics

Author: S. Lipschutz, M. Lipson And V. H. Patil

Publisher: Tata McGraw - Hill Education

Edition: 3

Year: 2009

ISBN: 9780070669123

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Set Theory

Scilab code Exa 1.8 inclusion exclusion principle

```
disp('To find:number of mathematics students taking atleast one of the languages French(F),German(G) and Russian(R)')
F=65; //number of students studying French
G=45;  // number of students studying German
R=42;  //number of students studying Russian
FandG=20;  //number of students studying French and German
FandR=25;  //number of students studying French and Russian
GandR=15;   //number of students studying German and Russian
FandGandR=8; //number of students studying French, German and Russian
 //By inclusion-exclusion principle
 ForGorR=F+G+R-FandG-FandR-GandR+FandGandR;
disp(ForGorR,'the number of students studying atleast one of the languages :')
```

Scilab code Exa 1.9 Inclusion exclusion principle

```
disp('In a college ,120 mathematics students can opt
   for either French(F),German(G) or Russian(R)')
n=120;  //total number of students
F=65;  //number of students studying French
G=45;  //number of students studying German
R=42;  //number of students studying Russian
FandG=20;  //number of students studying French and
   German
FandR=25;  //number of students studying French and
   Russian
GandR=15;  //number of students studying German and
   Russian
FandGandR=8;  //number of students studying French,
   German and Russian
disp('using inclusion-exclusion principle:')
ForGorR=F+G+R-FandG-FandR-GandR+FandGandR;
disp(ForGorR,'number of students studying French or
   German or Russian')
FGnR=FandG-FandGandR;
disp(FGnR,'number of students studying French and
   German but not Russian')
FRnG=FandR-FandGandR ;
disp(FRnG,'number of students studying French and
   Russian but not German')
GRnF=GandR-FandGandR ;
disp(GRnF,'number of students studying German and
   Russian but not French')
OF=F-FGnR-FandGandR-FRnG ;
disp(OF,'number of students studying  Only French')
OG=G-FGnR-FandGandR-GRnF;
disp(OG,'number of students studying  Only German')
OR=R-FRnG-FandGandR-GRnF;
disp(OR,'number of students studying  Only Russian')
k=n-ForGorR;
disp(k,'number of students not studying any of the
   languages')
```

Scilab code Exa 1.13 Power sets

```
x=10; //number of members of set X
P=2^x //number of members of the power set of X
q=P-1;//x itself is not the proper subset.Hence it
   isn't counted
disp(q,'number of members of powerset P which are
   proper subsets of x are:')
```

Scilab code Exa 1.14 Power sets

```
A=[1,2,3,4,5];  //eatables for salad preparation 1=
   onion,2=tomato,3=carrot,4=cabbage,5=cucumber
p=length(A);   //total number of eatables available
n=2^p-1;  //no salad can be made without atleast one
    of the eatables.Hence null set isn't counted
disp(n,'number of different salads that can be
   prepared using the given eatables')
```

Scilab code Exa 1.18 Mathematical induction

```
U1=1;  //given
U2=5;  //given
P=[];
for i=1:2
P(i)=3^i-2^i;
disp(P(i))
end
disp('P(1)=U(1) and P(2)=U(2)');
```

```
disp('hence Un=3^n-2^n for all n belonging to N');
```

Chapter 3

Functions and Algorithms

Scilab code Exa 3.8 Recursively defined functions

```
function[k]=fact(a)
k=-1;
if(a<0|a>200)
disp("Invalid");
break;
 else
if(a==1|a==0)
k=1;
 else
k=a*fact(a-1);
end
end
endfunction
a=4;
p=fact(a);
disp(p,'the value of 4! is')
```

Scilab code Exa 3.9 Cardinality

```
x=1;
y=2;
z=3;
A=[x,y,z];
disp('cardinality of set A is:')
length(A)
B=[1,3,5,7,9]
disp('cardinality of set B is:')
length(B)

// 3.9 (b)
disp('the set E has the following elements)
E=[2,4,6 %inf]  //set E is the set of all positive
    even numbers and N is the set of all natural
    numbers
disp('function f:N to E is defined.So,E has the same
     cardinality as N')
disp('set E is countably infinite:')
for x=2:2:%inf
y=2*x;
disp(y)
end
```

Scilab code Exa 3.10 Polynomial evaluation

```
x = poly(0, 'x');
p = 2*x^3-7*x^2+4*x-15;
disp(p,'the polynomial is')
k=horner(p,5);
disp(k,'value of the polynomial at x=5 is')
```

Scilab code Exa 3.11 Greatest Common Divisor

```
V=int32([258,60]);
thegcd=gcd(V);
disp(thegcd,'the gcd of the two numbers 258 and 60
   is')
```

Chapter 5

Vectors and Matrices

Scilab code Exa 5.2 Vector operations

```
u=[2,3,-4];
v=[1,-5,8];
u+v
5*u
-v
2*u-3*v
u.*v;
k=sum(u.*v);
disp(k,'dot product of the two vectors')
l=norm(u);
disp(l,'norm or length of the vector u')
```

Scilab code Exa 5.3 Column vectors

```
u=[5,3,-4]'
v=[3,-1,-2]'
2*u-3*v
k=sum(u.*v);
```

```
disp(k,'The dot product of the two vectors u and v
    is:')
l=norm(u);
disp(l,'The length or norm of the vector u is:')
```

Scilab code Exa 5.5 Matrix addition and Scalar multiplication

```
A=[1,-2,3;0,4,5];
B=[4,6,8;1,-3,-7];
k=A+B;
disp(k,'The addition of the two matrices A and B is:
    ')
m=3*A;
disp(m,'The multiplication of a vector with a scalar
     is:')
p=2*A-3*B
```

Scilab code Exa 5.6 Matrix multiplication

```
a=[7,-4,5];
b=[3,2,-1]';
k=a*b;
disp(k,'product of a and b is;')
p=[6,-1,8,3];
q=[4,-9,-2,5]';
l=p*q;
disp(l,'product of p and q is:')
```

Scilab code Exa 5.7 Matrix multiplication

```
A=[1 3;2 -1];
B=[2 0 -4;5 -2 6];
A*B
A=[1 2;3 4]
B=[5 6;0 -2];
A*B
B*A
disp('matrix mulitplication is not commutative since
    AB may not be equal to BA')
```

Scilab code Exa 5.8 Algebra of square matrices

```
A=[1 2;3 -4];
A2=A*A  //multiplying A by itself
A3=A2*A
f=2*A2-3*A+5;
disp(f,'for the function f(x)=2x^2-3x+5,f(A) is:')
g=A2+3*A-10;
disp(g,'for the function g(x)=x^2+3x-10,g(A) is')
```

Scilab code Exa 5.9 Invertible matrices

```
A=[1 0 2;2 -1 3;4 1 8];
B=[-11 2 2;-4 0 1;6 -1 -1];
A*B
disp('since A*B is identity matrix,A and B are
   invertible and inverse of each other')
```

Scilab code Exa 5.10 Determinants

```
A=[5 4;2 3];
det(A);
disp(det(A),'determinant of A')
B=[2 1;-4 6]
det(B);
disp(det(B),'determinant of B')
C=[2 1 3;4 6 -1;5 1 0]
disp(det(C),'determinant of C')
```

Scilab code Exa 5.13 Matrix solution of a system of linear equations

```
A=[1 2 1;2 5 -1;3 -2 -1];      //left hand side of the system of equations
B=[3 -4 5]';                    //right hand side or the constants in the equations
X=[];
X=A\B ;    //unique solution for the system of equations
x=X(1)
y=X(2)
z=X(3)
```

Scilab code Exa 5.14 Inverse of a square matrix

```
A=[1 0 2;2 -1 3;4 1 8];
P=rref([A,eye(3,3)]);
disp(P,'canonical form of matrix A :')
disp('left side of the matrix P is the identity matrix so the right side is the inverse of A')
inverseA=P(:,4:6)
```

Chapter 6

Counting

Scilab code Exa 6.1 Sum rule principle

```
M=8;     //number of male professors teaching
   calculus
F=5;     //number of female professors teaching
   calculus
T=M+F ;
disp(T,'number of ways a student can choose a
   calculus professor')

E=[2,3,5,7];    //event of choosing a prime number
   less than 10
F=[2,4,6,8];    //event of choosing an even number
   less than 10
G=intersect(E,F);  //event of getting an even and
   prime number
H=length(E)+length(F)-length(G);
disp(H,'event of getting an even or a prime number')

E=[11,13,17,19]; //event of choosing a prime number
   between 10 and 20
F=[12,14,16,18];  //event of choosing an even number
    between 10 and 20
```

```
G=union(E,F);     //event of choosing a number which is prime or even
k=length(G);
disp(k,'number of ways of choosing a number which is prime or even')
```

Scilab code Exa 6.2 Product rule principle

```
disp('a license plate contains two letters followed by three digits where first digit can not be zero')
n=26; //number of english letters
n*n;  //number of ways of choosing two letters in the license plate
p=10; //number of digits (0-9)
(p-1)*p*p; //number of ways to select the three digits with the first digit not being zero
k=n*n*(p-1)*p*p;
disp(k,'total number of license plates that can be printed')

disp('a president ,a secretary and a treasurer has to be elected in an orga-nisation of 26 members. No person is elected to more than one postion')
t=26;  //total number of members in the organisation
j=t*(t-1)*(t-2);
disp(j,'number of ways to elect the three officers (president,secretary,treasurer')
```

Scilab code Exa 6.3 Factorial notation

```
disp('To find: factorial of a 6')
facto2=2*1;
```

```
facto3=3*facto2
facto4=3*facto3
facto4=4*facto3
facto5=5*facto4
facto6=6*facto5

k=8*7*factorial(6)/factorial(6);
disp(k,'value of 8!/6! is:')
j=12*11*10*factorial(9)/factorial(9);
disp(j,'value of 12!/9! is:')
```

Scilab code Exa 6.4 Binomial coefficients

```
function [k]=func1(n,r)  //calculating binomial coefficient
k=factorial(n)/(factorial(r)*factorial(n-r));
endfunction
func1(8,2)
func1(9,4)
func1(12,5)
func1(10,3)
func1(13,1)

p = factorial(10)/(factorial(10-7)*factorial(7)); //calculating 10C7
q= factorial(10)/(factorial(10-3)*factorial(3))  //calculating 10C3
disp(p,'value of 10C7 is')
//10-7=3 so 10C7 can also be computed as 10C3
//both p and q have same values but second method saves time and space
```

Scilab code Exa 6.5 Permutations

```
disp('finding the number of three-letter words using
    only the given six letters(A,B,C,D,E,F) without
   repetition')
n=6;  //total number of letters
l1=n;  //number of ways in which first letter of the
    word can be chosen
l2=n-1; //number of ways in which second letter of
   the word can be chosen
l3= n-2; //number of ways in which third letter can
   be chosen
k=l1*l2*l3;
disp(k,'number of three-letter words  possible')
```

Scilab code Exa 6.6 Permutations with repetitions

```
function [k]= funct1(n,p,q)
k= factorial(n)/(factorial(p)*factorial(q));
endfunction
k=funct1(7,3,2)  //in "BENZENE" three letters are
   alike(the three Es) and two are alike (the two Ns
   )
disp(k,'The number of seven-letter words that can be
    formed using letters of the word BENZENE')

disp('a set of 4 indistinguishable red coloured
   flags , 3 indistinguishable white flags and a blue
    flag is given')
j=funct1(8,4,3);
disp(j,'number of different signals ,each consisting
    of eight flags')
```

Scilab code Exa 6.7 Combinations

```
disp('four objects are given (a,b,c,d) and three are taken at a time')
combinations = factorial(4)/(factorial(4-3)*factorial(3));
disp(combinations,'number of combinations of the four objects given')
k=factorial(3);  //number of permutations of objects in a combination
permutations = combinations*k;
disp(permutations,'total number of permuatations for the problem')
```

Scilab code Exa 6.8 Combinations

```
function [k]= myfunc(n,r)
k=factorial(n)/(factorial(n-r)*factorial(r));
endfunction
k=myfunc(8,3);
disp(k,'the number of committees of three that can be formed out of eight people')

cows=myfunc(6,3)  //number of ways that a farmer can choose 3 cows out of 6 cows
pigs=myfunc(5,2)  //number of ways that a farmer can choose 2 pigs out of 5 pigs
hens=myfunc(8,4)  //number of ways that a farmer can choose 4 hens out of 8 hens
p=cows*pigs*hens;
disp(p,'total number of ways that a farmer can choose all these animals')
```

Scilab code Exa 6.9 Combinations with repetitions

```
//each solution  to the equation can be viewed as a
   combination of objects
r=18;  //number of objects
M=3;   //kinds of object
m=factorial(r+(M-1))/(factorial(r+(M-1)-(M-1))*
   factorial(M-1));
disp(m,'number of non negative integer solutions of
   the given equation x+y+z=18')
```

Scilab code Exa 6.14 Ordered partitions

```
c1=3; //number of toys that the youngest child
   should get
c2=2;  //number of toys that  the third child should
    get
c3=2;  //number of toys that the second child should
    get
c4=2;   //number of toys that the eldest son should
   get
m=factorial(9)/(factorial(3)*factorial(2)*factorial
   (2)*factorial(2));
disp(m,'number of ways nine toys can be divided
   between four children with the youngest son
   getting 3 toys and others getting 2 each')
```

Scilab code Exa 6.15 Unordered partitions

```
p=12;  //total number of students
t=3;  //number of teams or partition
disp('each partition of the students can be arranged
    in 3! ways as an ordered partition')
r=factorial(12)/(factorial(4)*factorial(4)*factorial
   (4))  //number of ordered partitions
```

```
m=r/factorial(t);  //number of unordered partitions
disp(m,'number of ways that 12 students can be
   partitioned into three teams so that each team
   consists of 4 students')
```

Scilab code Exa 6.16 Inclusion exclusion principle revisited

```
U=1000;  //number of elements in the set of positive
    integers not exceeding 1000
A=U/3;   //number of elements in the subset of
   integers divisible by 3
B=U/5;   //number of elements in the subset of
   integers divisible by 5
C=U/7;   //number of elements in the subset of
   integers divisible by 7
AandB=floor(U/(3*5))   //number of elements in the
   subset containing numbers divisible by both 3 and
    5
AandC=floor(U/(3*7))   //number of elements in the
   subset containing numbers divisible by both 3 and
    7
BandC=floor(U/(5*7))   //number of elements in the
   subset containing numbers divisible by both 5 and
    7
AandBandC=floor(U/(3*5*7))  //number of elements in
   the subset containing numbers divisible by 3,5
   and 7
s=U-(A+B+C)+(AandB+AandC+BandC)-(AandBandC);  // By
   inclusion-exclusion principle
S=round(s);
disp(S,'The number of integers in the set U, which
   are not divisible by 3,5 and 7 is')
```

Chapter 7

Probability Theory

Scilab code Exa 7.1 Sample space and events

```
S=[1,2,3,4,5,6];          //sample space for the
    rolling of a die
A=[2,4,6];                //event that an even number
    occurs
B=[1,3,5];               //event that an odd number
    occurs
C=[2,3,5];              //event that a prime number
    occurs
disp(union(A,C),'sample space for the event that an
    even or a prime number occurs')
disp(intersect(B,C),'sample space for the event that
     an odd prime number occurs')
disp(setdiff(S,C),'sample space for the event that a
     prime number does not occur')      //It is the
    complement of the set C.
intersect(A,B)      //It is a null set or null vector
     since there can't occur an even and an odd
    number simultaneously

H=0;                  //"head" face of a coin
T=1;                 //"tail" face of a coin
```

```
S=["000","001","010","011","100","101","110","111"];     //sample space for the toss of a coin three times
A=["000","001","100"];        //event that two more or more heads appear consecutively
B=["000","111"];              //event that all tosses are the same
disp(intersect(A,B),'sample space for the event in which only heads appear')

disp('Experiment:tossing a coin until a head appears and then counting the number of times the coin is tossed')
S=[1,2,3,4,5,%inf]         //The sample space has infinite elements in it
disp("Since every positive integer is an element of S,the sample space is infinite")
```

Scilab code Exa 7.2 Finite probability spaces

```
disp('Experiment:three coins are tossed and the number of heads are observed')
S=[0,1,2,3];    //the sample space for the experiment where 0 implies no heads,1 implies only one head out of the three coins and so on
disp("the probability space is as follows ")
P0=1/8;  //probability of getting no head on any of the coins i.e TTT
P1=3/8;  //probability of getting only one head on any of the coins, out of the three coins i.e HTT,THT,TTH
P2=3/8;  //probability of getting two heads, out of the three coins i.e THH,HTH,HHT
P3=1/8;   //probability of getting all the three heads i.e HHH
```

```
disp("A is the event that atleast one head appears and B is the event that all heads or all tails appear ")
A=[1,2,3];    // HHH,HHT,HTH,HTT,THH,THT,TTH
B=[0,3];    //HHH,TTT
PA=P1+P2+P3;
disp(PA,'probability of occurrence of event A')
PB=P0+P3;
disp(PB,'probability of occurrence of event B')
```

Scilab code Exa 7.3 Equiprobable spaces

```
disp("Experiment: a card is selected from a deck of 52 cards ")
disp("A is the event of the selected card being a spade ")
disp("B is the event of the selected card being a face card ")
t=52 ;       //the total number of cards
s=13;       //number of spades
PA= s/t;
disp(PA,'probability of selecting a spade')
f=12;      //number of face cards(jack,queen,king)
PB=f/t;
disp(PB,'probability of selecting a face card')
sf=3;     //number of spade face cards
Psf=sf/t;
disp(Psf,"probability of selecting a spade face card is:")
```

Scilab code Exa 7.4 Addition principle

```
disp("Experiment: selection of a student out of 100
    students ")
M=30;   //no of students taking mathematics
C=20;   //no of students taking chemistry
T=100;  //total no. of students
PM = M/T  //probability of the selected student
    taking mathematics
PC = C/T  //probability of the selected student
    taking chemistry
MnC=10;  //no of students taking mathematics and
    chemistry
PMnC = MnC/T   //probability of the selected student
     taking mathematics and chemistry both
PMorC = PM+PC-PMnC;
disp(PMorC,'probability of the selected student
    taking mathematics or chemistry')
```

Scilab code Exa 7.5 Conditional probability

```


//EXAMPLE 7.5 (a)


 disp(" Experiment: A die is tossed and the outcomes
     are observed");


 disp("To find: probability (PM) of an event that
     one of the dice is 2 if the sum is 6");


E=["(1,5)","(2,4)","(3,3)","(4,2)","(5,1)"]     //
    event that the sum of the two numbers on the two
    dice is 6
```

```
A=["(2,1)","(2,2)","(2,3)","(2,4)","(2,5)","(2,6)","(1,2)","(3,2)","(4,2)","(5,2)","(6,2)"]  //event that 2 appears on atleast one die


B= intersect(A,E)    //possible combination of numbers on two die such that their sum is 6 and 2 appears atleast on one die


PM=2/5  //since E has 5 elements and B has 2 elements






//EXAMPLE 7.5(b)

disp("A couple has two children");


b=1;    //boy child

g=2;    //girl child

S=[11,12,21,22] ;     //sample space where 11 implies both children being boys,12 implies first child being a boy and the second child being a girl and so on

disp("To find: probability(PM) that both children are boys  ");



```

```
//7.5(b).i

L=S(:,1:3)     //reduced sample space if it is known
    that one of the children is a boy


PM=1/length(L)


//7.5(b).ii

R=S(:,1:2)     //reduced sample space if it is known
   that the older child is a boy


PM=1/length(R)
```

Scilab code Exa 7.6 Multiplication theorem for conditional probability

```
disp("A bag contains 12 items of which four are
   defective.Three items are drawn at random,one
   after the other");
s=12;   //total itmes in the bag
d=4;  //defective items in the bag
Pf=(s-d)/s ;  //probability that the first item
   drawn is  non defective
Pe=Pf*[(s-d-1)/(s-1)]*[(s-d-2)/(s-2)];
disp(Pe,'probability that all three items are non
   defective')
//after the first item is chosen,the second item is
   to be chosen from 1 less than the original number
    of items in the box and similarly the number of
   non defective items gets decreased by 1.Similarly
    ,for the third draw of item from the box
```

Scilab code Exa 7.7 Independent events

```
H=1;    //heads of a coin
T=2;    //tails of the coin
S=[111,112,121,122,211,212,221,222]  //sample space for the toss of a coin three times. 111 implies heads all three times,112 implies heads on first two tosses and tails on the third toss
A=[111,112,121,122];  //event that first toss is heads
B=[111,112,211,212];  //event that second toss is heads
C=[112,211];          //event that exactly two heads appear in a row
PA=length(A)/length(S);
disp(PA,'probability of A is')
PB=length(B)/length(S);
disp(PB,'probability of B is')
PC=length(C)/length(S);
disp(PC,'probability of C is')
AnB=intersect(A,B)
AnC=intersect(A,C)
BnC=intersect(B,C)
PAnB= length(AnB)/length(S);
disp(PAnB,'probability of the event AnB')
PAnC= length(AnC)/length(S);
disp(PAnC,'probability of the event AnC')
PBnC= length(BnC)/length(S);
disp(PBnC,'probability of the event BnC')
if((PA*PB)==PAnB),
 disp("A and B are independent")
else
 disp("A and B are dependent")
end
```

```
if((PA*PC)==PAnC),
  disp("A and C are independent")
else
 disp("A and C are dependent")
end
if((PB*PC)==PBnC),
  disp("B and C are independent")
else
  disp("B and C are dependent")
end
```

Scilab code Exa 7.8 Independent events

```
disp("Experiment: A and B both shoot at a target")
PA=1/4;  //given probability of A hitting the target
PB=2/5;  //given probability of B hitting the target
disp("A and B are independent events so PA*PB will
    be equal to probability of the event of A and B
    both hitting the target i.e PAnB")
PAnB=PA*PB;
PAorB=PA+PB-PAnB;
disp(PAorB,'probability of atleast one of them
    hitting the target')
```

Scilab code Exa 7.9 Independent repeated trials

```
disp("Experiment: Three horses race together twice")
Ph1=1/2;  //probability of first horse winning the
    race
Ph2=1/3;   //probability of second horse winning the
     race
Ph3=1/6;   //probability of third horse winning the
    race
```

```
S=[11,12,13,21,22,23,31,32,33]  //sample space where
     11 implies first horse winning the first and
    second race both,12 implies first horse winning
    the first race and second horse winning the
    second race and so on
P11=Ph1*Ph1  //probability of first horse winning
    both races
P12=Ph1*Ph2  //probability of first horse winning
    the first race and second horse winning the
    second race
P13=Ph1*Ph3  //probability of first horse winning
    the first race and third horse winning the second
     race
P21=Ph2*Ph1  //probability of second horse winning
    the first race and first horse winning the second
     race
P22=Ph2*Ph2  //probability of second horse winning
    both the races
P23=Ph2*Ph3  //probability of second horse winning
    the first race and third horse winning the second
     race
P31=Ph3*Ph1  //probability of third horse winning
    the first race and first horse winning the second
     race
P32=Ph3*Ph2  //probability of third horse winning
    the first race and second horse winning the
    second race
P33=Ph3*Ph3  //probability of third horse winning
    both the races
disp(P31,'probability of third horse winning the
    first race and first horse winning the second
    race is')
```

Scilab code Exa 7.10 Repeated trials with two outcomes

```
n=6;    //number of times a fair coin is tossed and
    getting a heads is a success
p=1/2;  //probability of getting a heads
q=1/2 ; //probability of not getting a heads
P2=(factorial(6)/(factorial(6-2)*factorial(2)))*p^2*
    q^(6-2);
disp(P2,'probability of getting exactly two heads (i
    .e k=2)')

P4=(factorial(6)/(factorial(6-4)*factorial(4)))*p^4*
    q^(6-4); //probabilty of getting four heads
P5=(factorial(6)/(factorial(6-5)*factorial(5)))*p^5*
    q^(6-5);  //probabilty of getting five heads
P6=(factorial(6)/(factorial(6-6)*factorial(6)))*p^6*
    q^(6-6);  //probabilty of getting five heads
PA=P4+P5+P6 ;
disp(PA,'probability of getting atleast four heads(i
    .e k=4,5 or 6)')

Pn=q^6          //probability of getting no heads
Pm=1-Pn;
disp(Pm,'probability of getting one or more heads')
```

Scilab code Exa 7.12 Random variables

```
disp("A box contains 12 items of which three are
    defective")
disp("A sample of three items is selected from the
    box")
s=factorial(12)/(factorial(12-3)*factorial(3));
disp(s,'number of elements in the sample space where
     samples are of size 3')
//X denotes the number of defective items in the
    sample
x=[0,1,2,3];  //range space of the random variable X
```

Scilab code Exa 7.13 Probability distribution of a random variable

```
r=[1,2,3,4,5,6,5,4,3,2,1];
//number of outcomes whose sum is 2,3,4,5,6,7,8,9,10,11,12 respectively such that there is only 1 outcome i.e (1,1) whose sum is 2, two outcomes (1,2) and (2,1) whose sum is 3 and so on
t=36;                                   //total number of elements in the sample space of the experiment of tossing a pair of dice
for i=1:11;
p=r(i)/t;
disp(p)
end
0.0277778                     //probability of getting a sum of 2
0.0555556                     //probability of getting a sum of 3
0.0833333                     //probability of getting a sum of 4
0.1111111                     //probability of getting a sum of 5
0.1388889                     //probability of getting a sum of 6
0.1666667                     //probability of getting a sum of 7
0.1388889                     //probability of getting a sum of 8
0.1111111                     //probability of getting a sum of 9
0.0833333                     //probability of getting a sum of 10
0.0555556                     //probability of getting a sum
```

```
  of 11
0.0277778            //probability of getting a sum
  of 12
x=[2,3,4,5,6,7,8,9,10,11,12];       //range space of
  random variable X which assigns to each point in
  sample space the sum of the numbers
D=[ 2,3,4,5,6,7,8,9,10,11,12;  0.0277778, 0.0555556
  , 0.0833333,  0.1111111, 0.1388889 ,0.1666667,
  0.1388889 ,0.1111111, 0.0833333, 0.0555556,
  0.0277778];
disp(D,'distribution table of X where first row
  gives the range space and second row gives the
  respective probabilities is as follows')
```

Scilab code Exa 7.14 Probability distribution of a random variable

```
disp("a box contains 12 items of which three are
  defective")
disp("A sample of three items is selected from the
  box")
r=factorial(9)/(factorial(9-3)*factorial(3))  //
  number of samples of size 3 with no defective
  items
t=220;                                       //
  number of different samples of size 3 i.e the
  number of elements in the sample space
P0=r/t                                      //
  probability of getting no defective item
r1=3*(factorial(9)/(factorial(9-2)*factorial(2)))
        //number of samples of size 3 getting 1
  defective item
P1=r1/t

  //probability of getting 1 defective item
r2=9*(factorial(3)/(factorial(3-2)*factorial(2)))
```

```
            //number of samples of size 3 getting 2 defective item
P2=r2/t                                                                                //probability of getting 2 defective item
r3=1;                          //number of samples of size 3 getting 3 defective item
P3=r3/t                        //probability of getting 3 defective item
x=[0,1,2,3];
p=[P0,P1,P2,P3];
D=[0,1,2,3;P0,P1,P2,P3];
disp(D,'distribution table for random variable X the upper row being values of X')
```

Scilab code Exa 7.15 Expectation of a random variable

```
disp("A fair coin is tossed six times");
x=[0,1,2,3,4,5,6];    //number of heads which can occur
p=[1/64,6/64,15/64,20/64,15/64,6/64,1/64];   //probability of occurring of heads where 1/64 is probability for occurrence of a single head,6/64 that of occurrence of two heads and so on.
r=0;
for i=1:7;
r = r + (x(i)*p(i));
end
disp(r,'mean or expected number of heads are')

disp("X is a random variable which gives possible number of defective items in a sample of size 3");
//Box contains 12 items of which three are defective
x=[0,1,2,3];    //possible number of defective items
```

```
in a smaple of size 3
p=[84/220,108/220,27/220,1/220];  //probability of occurrence of each number in x respectively where 84/220 is the probability for getting no defective item,108/220 is that of getting 1 defective item and so on.
 r=0;
for i=1:4;
r = r + (x(i)*p(i));
end
disp(r,'expected number of defective items in a sample of size 3 are')

Ph1=1/2;    //probability of winning the race by first horse
Ph2=1/3;    //probability of winning the race by second horse
Ph3=1/6;    //probability of winning the race by third horse
//X is the payoff function for the winning horse
X1=2;      //X pays $2 as first horse wins the rac
X2=6;      //X pays $6 as second horse wins the race
X3=9;      //X pays $9 as third horse wins the race
E=X1*Ph1+X2*Ph2+X3*Ph3;
disp(E,'expected pay off for the race is')
```

Scilab code Exa 7.16 Variance and standard deviation of a random variable

```
u=3;                //mean of distribution of random variable X
x=[0,1,2,3,4,5,6];   //values of X in the distribution as x where it is the number of times heads occurs when a coin is tossed six times
p=[1/64,6/64,15/64,20/64,15/64,6/64,1/64];  //
```

```
    probabilities of occurrence of each value of X (x
    ) in the distribution such that 1/64 gives the
    probability of occurrence of no heads at all,6/64
     gives that of occurrence of heads for only one
    time and so on
k=0;
for i=1:7;
k=k+((x(i)-u)^2)*p(i);
end
disp(k,'Variance of X is')
s=sqrt(k);
disp(s,'Standard deviation of X is')

u=0.75;    //mean
x=[0,1,2,3];  //values of random variable X as x in
    the probability distribution of X
p=[84/220,108/220,27/220,1/220];   //probability of
    values in x which appear in distribution table of
     X
g=0;
for i=1:4;
g=g+((x(i))^2)*p(i);
end
h=g-(u*u);
disp(h,'variance of X is')
sd=sqrt(h);
disp(sd,'Standard deviation for X')
```

Scilab code Exa 7.17 Binomial diatribution

```
p=1/5;  //probability of the man hitting a target
q=1-1/5; //probability of the man not hitting the
    target
n=100;  //number of times the man fires
e=n*p;
```

```
disp(e,'expected number of times the man will hit
   the target')
r=sqrt(n*p*q);
disp(r,'Standard deviation')

p=1/2;  //probability of guessing the correct answer
    in a five question true-false exam
n=5;   //number of questions in the exam
g=n*p;
disp(g,'expected number of correct answers in the
   exam')
```

Scilab code Exa 7.18 Chebyshev inequality

```
u=75;     //mean of a random variable X
n=5;      //standard deviation of X
k=2;          //for k=2
l1=u-k*n
l2=u+k*n
P1=1-(1/k)^2
disp("thus the probability that a value of X lies
   between 65 and 85 is atleast 0.75 according to
   Chebyshev inequality")
k=3;                //for k=3
l1=u-k*n
l2=u+k*n
P2=1-(1/k)^2
disp("thus the probability that a value of X lies
   between 60 and 90 is atleast 0.8888889 according
   to Chebyshev Inequality")
```

Scilab code Exa 7.19 Sample mean and Law of large numbers

```
disp(" a die is tossed 5 times with the following outcomes")
x1=3;
x2=4;
x3=6;
x4=1;
x5=4;
xmean=(x1+x2+x3+x4+x5)/5   //mean of the outcomes
disp('for a fair die the mean is 3.5.So law of large numbers tells us that as number of outcomes increase for this experiment,there is a greater likelihood that themean will get closer to 3.5')
```

Chapter 8

Graph Theory

Scilab code Exa 8.1 Paths and connectivity

```
// refer to page 8.6
disp('given a graph with 6 nodes viz. node1,node2
    ....node6')
A=[0 1 0 1 1 0;1 0 1 0 1 0;0 1 0 0 0 1;1 0 0 0 0 0;1
     1 0 0 0 0;0 0 1 0 0 0];
disp(A,'The adjacency matrix for A is')
disp('sequence A is a path from node4 to node6; but
    it is not a trail since the edge from node1 to
    node2 is used twice')
B=[0 0 0 1 1 0;0 0 0 0 1 1;0 0 0 0 0 0;1 0 0 0 0 0;1
     1 0 0 0 0;0 1 0 0 0 0];
disp(B,'The adjacency matrix for B is')
disp('sequence B is not a path since there is no
    edge from node2 to node6 is used twice')
C=[0 0 0 1 1 0;0 0 1 0 1 0;0 1 0 0 1 0;1 0 0 0 0 0;1
     1 1 0 0 1;0 0 0 0 1 0];
disp(C,'The adjacency matrix for C is')
disp('sequence C is a trail since is no edge is used
     twice')
D=[0 0 0 1 1 0;0 0 0 0 0 0;0 0 0 0 1 1;1 0 0 0 0 0;1
     0 1 0 0 0;0 0 1 0 0 0];
```

```
disp(D,'The adjacency matrix for D is')
disp('sequence D is a simple path from node4 to
    node6')
```

Scilab code Exa 8.2 Minimum spanning tree

```
disp('to find:minimal spanning tree')
disp('the adjacency matrix for the weighted graph(
    nodeA,nodeB...nodeF) of 6 nodes is :')
K=[0 0 7 0 4 7;0 0 8 3 7 5;7 8 0 0 6 0;0 3 0 0 0 4;4
     7 6 0 0 0;7 5 0 4 0 0]
disp('edges of the graph')
AC=7;
AE=4;
AF=7;
BC=8;
BD=3;
BE=7;
BF=5;
CE=6;
DF=4;
M=[AC,AE,AF,BC,BD,BE,BF,CE,DF];  //set of all edges
V=int32(M);
L=gsort(V) //edges sorted in decreasing order of
    their weights
disp('deleting edges without disconnecting the graph
     until 5 edges remain')
N=[BE,CE,AE,DF,BD];  //edges in minimum spanning
    tree
Sum=sum(N);
disp(Sum,'weight of the minimal spanning tree is')


disp('another method of finding a minimal spanning
    tree is :')
```

```
K=gsort(V,'g','i')    //edges sorted in increasing order
N2=[BD,AE,DF,CE,AF];   //edges in minimum spanning tree
Sum2=sum(N2);
disp(Sum2,'weight of the minimal spanning tree is')
```

Chapter 9

Directed graphs

Scilab code Exa 9.6 Adjacency matrix

```
A=[0 0 0 1;1 0 1 1;1 0 0 1;1 0 1 0];
disp(A,'adjacency matrix of graph G is')
A2=A^2
A3=A^3
disp('the number of ones in A is equal to the number
    of edges  in the graph i.e 8')
```

Scilab code Exa 9.8 Path matrix

```
A=[0 0 0 1;1 0 1 1;1 0 0 1;1 0 1 0];
disp(A,'adjacency matrix of graph G is')
A4=A^4;
A3=A^3;
A2=A^2;
B4=A+A2+A3+A4;
B4=[4 11 7 7 0 0 0 0 3 7 4 4 4 11 7 7];
for i=1:16
if(B4(i)~=0) then
```

```
B4(i)=1;
end
end
disp(B4,'Replacing non zero entries of B4 with 1 ,we
    get path (reachability) matrix P is:')
disp('there are zero entries in P,therefore the
   graph is not strongly connected')
```

Chapter 11

Properties of the integers

Scilab code Exa 11.2 Division algorithm

```
disp('Division Algorithm')
a=4461;   //dividend
b=16;     //divisor
r=modulo(a,b)  //remainder
k=fix(a/b)     //quotient
j=b*k+r   //dividend=divisor*quotient+remainder

a=-262;   //dividend
b=3;     //divisor
k=fix(a/b)  //remainder
r=modulo(a,b)     //quotient
j=b*k+r   //dividend=divisor*quotient+remainder
disp('a and j have equal values.Hence division
   algorithm is proved')
```

Scilab code Exa 11.4 Primes

```
disp('Divisibility and Primes')
```

```
x=50;
disp('prime numbers less than 50 are')
y=primes(x)

disp('the prime factorisation of 21,24 and 1729
   respectively are:')
k=factor(21)
l=factor(24)
n=factor(1729)
```

Scilab code Exa 11.5 Greatest Common Divisor

```
disp('the GCD of the following numbers is:')
V=int32([12,18]);
[thegcd]=gcd(V)
V=int32([12,-18]);
[thegcd]=gcd(V)
V=int32([12,-16]);
[thegcd]=gcd(V)
V=int32([29,15]);
[thegcd]=gcd(V)
V=int32([14,49]);
[thegcd]=gcd(V)
```

Scilab code Exa 11.6 Euclidean algorithm

```
disp('Euclidean Algorithm')
a=[540,168,36,24];
b=[168,36,24,12];
for i=1:4
V=int32([a(i),b(i)]);
thegcd=[];
thegcd(i)=gcd(V);
```

```
disp(thegcd(i))
end

function []=myf(dividend,divisor)
quotient=floor(dividend/divisor);
rem=modulo(dividend,divisor);
k=quotient*divisor+rem;
disp(k)
if(rem~=0) then
     myf(divisor,rem)
end
endfunction

myf(540,168)

disp('for the equation 540*x+168*y=12,we are given')
a=540;
b=168;
c=24;
d=36;
d=a-3*b;     //Eqn (1)
c=b-4*d;      //Eqn (2)
k=d-1*c;    //Eqn (3)
5*d-1*b;      //Eqn (4)
k=d-b+4*d;     //substituting value of c in Eqn (3)
   from Eqn (2)
r=5*a-16*b;
if(r==k) then
    disp('x=5 and y=16');
end
```

Scilab code Exa 11.9 Fundamental theorem of Arithmetic

```
a=2^4*3^3*7*11*13
b=2^3*3^2*5^2*11*17
```

```
V=int32([a,b]);
[d]=gcd(V)
lcm1=2^4*3^3*5^2*7*11*13*17    //lcm is the product
   of those primes which appear in either a or b
```

Scilab code Exa 11.12 Congruence relation

```
x=poly(0,'x');
g=3*x^2-7*x+5
m=horner(g,2)  //value of polynomial at 2
n=horner(g,8)  //value of polynomial at 8
j=m-n
disp(n,"for n = ")
if(modulo(j,6)==0) then
mprintf('%i is congruent to %i(mod 6)',m,n)
end
```

Scilab code Exa 11.19 Linear congruence equation

```
disp('solving for the congruence equation 8x @ 12(
   mod 28) ,where @ is the sign for congruence')
a=8;
b=12;
m=28;
V= int32([a,m]);
d= gcd(V);
a1= a/d;
b1= b/d;
m1= m/d;

function yd= f(x)
yd= (a1*x)-b1
endfunction
```

```

 disp('k is the unique solution of the equation ')
for i=0 : m1
x=i;
p=f(x);
if(modulo(p,m1) == 0)
k=x
break;
end
end

s1=k;
s2=k+m1;
s3=k+(m1*2);
s4=k+(m1*3);
 disp('solutions of the original equation at d=4')
disp(s1)
disp(s2)
disp(s3)
disp(s4)
```

Chapter 12

Algebraic Systems

Scilab code Exa 12.4 Properties of operations

```
a=(8-4)-3
b=8-(4-3)
disp('since a and b are not equal so subtraction is
   non-commutative on Z(set of integers)')

a=[1 2;3 4]
b=[5 6;0 -2]
g= a*b
k= b*a
disp('since g and k are not equal matrix
   multiplication is non-commutative')

h=(2^2)^3
j=2^(2^3)
disp('since h and j are not equal so exponential
   operation is non associative on the set of
   positive integers N')
```

Scilab code Exa 12.17 Roots of polynomial

```
t=poly(0,'t');
f=t^3+t^2-8*t+4
g=factors(f)
disp(r=roots(f),'roots of f(t) are as follows:')

t=poly(0,'t');
h=t^4-2*t^3+11*t-10
disp(r=roots(h),'the real roots of h(t) are 1 and −2
    ')
```

Scilab code Exa 12.18 Roots of polynomial

```
t=poly(0,'t');
f=t^4-3*t^3+6*t^2+25*t-39
g=factors(f)
 disp(r=roots(f),'roots of f(t) are as follows:')
```

Chapter 15

Boolean Algebra

Scilab code Exa 15.1 Basic definitions in boolean algebra

```
//0 denotes False and 1 denotes true
b=[0,1];
//binary operation  + on the set of bits
for i=1:2
for j=1:2
k = b(i)& b(j);
disp(k)
end
end
//binary operation  * on the set of bits
for i=1:2
for j=1:2
k = b(i)| b(j);
disp(k)
end
end
//unary operation ' on the set of bits
k=~b
clear;
D=[1,2,5,7,10,14,35,70];
a=35;
```

```
b=70;
V=int32([a,b]);
thelcm=lcm(V)   //a+b=lcm(a,b)
V=int32([a,b])
thegcd=gcd(V)   //a*b=gcd(a,b)
abar=70/a       //a'=70/a
```

Scilab code Exa 15.2 Boolean algebra as lattices

```
D=[1,2,5,7,10,14,35,70];
a = 2;  //a and b belong to D
b = 14;
V=int32([a,b]);
thelcm=lcm(V)
V=int32([a,b]);
thegcd=gcd(V)
abar=70/a
bbar=70/b
j=[abar,b];
h=[a,bbar];
V=int32([j])
lcm1=lcm(V)
K=int32([h])
lcm2=lcm(K)
```

Chapter 16

Recurrence relations

Scilab code Exa 16.14 Linear homogenous recurrence relations with constant coefficients

```
a=[];
a(1)=1;  //initial condition
a(2)=2;  //initial condition
 disp('for recurrence relation a(n)=5*a(n-1)-4*a(n
    -2)+n^2') //this is a second order recurrence
    relation with constant coefficients.It is non
    homogenous because of the n^2
for n=3:4
a(n)=5*a(n-1)-4*a(n-2)+n^2;
mprintf('Value of a(%i) is: %i \n',n,a(n))
end

a=[];
a(1)=1;   //initial condition
a(2)=2;   //initial condition
disp('for recurrence relation a(n)=2*a(n-1)*a(n-2)+n
   ^2')  //this recurrence relation is not linear
for n=3:4
a(n)=2*a(n-1)*a(n-2)+n^2;
mprintf('Value of a(%i) is: %i \n',n,a(n))
```

```
end

a=[];
a(1)=1;     //initial condition
a(2)=2;     //initial condition
disp('for recurrence relation a(n)=n*a(n-1)+3*a(n-2)')   //this is a homogenous linear second order recurrence relation without constant coefficients because the coefficient of a[n-1] is n,not a constant
for n=3:4
a(n)=n*a(n-1)+3*a(n-2);
mprintf('Value of a(%i) is: %i \n',n,a(n))
end


a=[];
a(1)=1;       //initial condition
a(2)=2;       //initial condition
a(3)=1;       //initial condition
disp('for recurrence relation a(n)=2*a(n-1)+5*a(n-2)-6*a(n-3)') //this is a homogenous linear third order recurrence relation with constant coefficients.Thus we need three,not two,initial conditions to yield a unique solution of the recurrence relation
for n=4:6
a(n)=2*a(n-1)+5*a(n-2)-6*a(n-3);
mprintf('Value of a(%i) is: %i \n',n,a(n))
end
```

Scilab code Exa 16.15 Solving linear homogenous recurrence relations with constant coefficients

```
disp('recurrence relation of Fibonacci numbers f[n]=
```

```
    f[n-1]+f[n-2]')
x=poly(0,'x');
g=x^2-x-1;
disp(g,'characterstic equation of the recurrence
    relation is:')
j=[];
j=roots(g);
disp(j,'roots of the characterstic equation j1,j2')
disp('for general equation fn=Ar^n+Br^n,values of
    Aand B respectively are calculated as:')
disp('initial condition at n=0 and n=1 respectively
    are:')
f1=1;
f2=1;
//putting the values of f1 and f2 we get the
    equations to solve
D=[ 1.6180340 -0.618034;(1.6180340)^2  (-0.618034)
    ^2];
K=[1 1]';
c=[];
c=D\K;
A=c(1)
B=c(2)

disp('thus the solution is f[n
    ]=0.4472136*((1.618034)^n-(- 0.4472136)^n)]')
```

Scilab code Exa 16.16 Solving linear homogenous recurrence relations with constant coefficients

```
disp('The recurrence relation t[n]=3t[n-1]+4t[n-2]')
x=poly(0,'x');
g=x^2-3*x-4;
disp(g,'characterstic polynomial equation for the
    above recurrence relation')
```

```
j=[];
j=roots(g);
disp(j,'roots of the characterstic equation j1,j2')
disp('general solution t[n]=c1*(-1)^n+c2*4^n)')
disp('initial condition at n=0 and n=1 respectively are:')
t0=0;
t1=5;
//putting the values of t0 and t1 we get the equations to solve
D=[1 1;-1 4]
K=[0 5]'
c=[];
c=D\K;
c1=c(1)
c2=c(2)
disp('thus the solution is t{n}=4^n-(-1)^n')
```

Scilab code Exa 16.17 Solving linear homogenous recurrence relations with constant coefficients

```
disp('The recurrence relation t[n]=4(t[n-1]-t[n-2])')
x=poly(0,'x');
disp(g=x^2-4*x+4,'characterstic polynomial equation for the above recurrence relation')
j=[];
j=roots(g);
disp(j,'roots of the characterstic equation j1,j2')
disp('the general solution is t[n]=n*2^n)
disp('initial condition at n=0 and n=1 respectively are:')
t0=1;
t1=1;
//putting the values of t0 and t1 we get the
```

```
equations to solve
D=[1 0;2 2]
K=[1 1]'
c=linsolve(D,K)
D=[1 0;2 2]
K=[1 1]'
c=[];
c=D\K;
c1=c(1)
c2=c(2)
disp('thus the solution is t{n}=2*n-n*2^(n-1)')
```

Scilab code Exa 16.18 Solving linear homogenous recurrence relations with constant coefficients

```
disp('The recurrence relation a[n]=2*a[n-1]-3a[n-2]')
x=poly(0,'x');
disp(g=x^2-2*x-3,'characterstic polynomial equation for the above recurrence relation')
j=[];
j=roots(g);
disp(j,'roots of the characterstic equation j1,j2')
disp('the general solution is a[n]=c1*3^n+c2*(-1)^n')
disp('initial condition at n=0 and n=1 respectively are:')
//putting the values of t0 and t1 we get the equations to solve
a0=1;
a1=2;
D=[1 1;3 -1]
K=[1 2]'
c=[];
c=D\K;
```

```
c1=c(1)
c2=c(2)
disp('thus the solution is a[n]=0.75*(3^n)+0.25*(1^n)')
```

Scilab code Exa 16.19 Solving general homogenous linear recurrence relations

```
disp('The recurrence relation a[n]=11*a[n-1]-39*a[n-2]+45*a[n-3]')
x=poly(0,'x');
disp(g=x^3-11*x^2+39*x-45,'characterstic polynomial equation for the above recurrence relation')
j=[];
j=roots(g);
disp(j,'roots of the characterstic equation j1,j2')
disp('hence the general solution is:a[n]=c1*(3^n)+c2*n*(3^n)+c3*(5^n)')
disp('initial condition at n=0 and n=1 respectively are:')
//putting the values of t0 and t1 we get the equations to solve
a0=5;
a1=11;
a2=25;
D=[1 0 1;3 3 5;9 18 25];
K=[5 11 25]'
c=[];
c=D\K;
c1=c(1)
c2=c(2)
c3=c(3)
disp('thus the solution is a[n]=(4-2*n)*(3^n)+5^n')
```