26/12/22

Design and Analysis of Algorithms (DAA)

Syllabus

UNIT 1 : Introduction. Divide and conquer

Introduction : Algorithm Definition, Algorithm specification, performance Analysis, performance Measurement, Asymptotic notations.

Divide and Conquer : General Method, Binary Search, finding the maximum and minimum. Quick sort.

UNIT - II : The Greedy Method

The Greedy Method : The General method, fractional knapsack problem, single source shortest path problem, optimal storage on Tapes problem, optimal Merge patterns problems.

UNIT - III : Dynamic programming

Dynamic programming : The General Method, 0/1 knapsack problem, single source shortest path - General weights, All Pairs - shortest paths problem, Traveling salesperson problem string editing problem.

UNIT - IV : Backtracking

Backtracking : The General method, the N-Queens problem, sum of subsets problem, Graph coloring problem, Hamiltonian cycles problem.

UNIT - V : Branch & Bound. NP-Hard and NP-Complete.

Branch & Bound : The General method, FIFO Branch-and-Bound, LC Branch-and-Bound, 0/1 knapsack problem, Travelling salesperson problem.

NP-Hard and NP-Complete problems : Basic Concepts, Cook's theorem.

24/12/22

1. Design and Analysis of Algorithms

UNIT - I

Def :-

Algorithm :-

The algorithm is a simply set of rules used to perform some calculations either by hand or most of the time use machines.

(or)

Set of instructions to solve a problem.

The algorithm consists of following characteristics

(1) Input (Zero or more inputs supplied to alg)

(2) output (one or more) Atleast one quantity is produced by every alg as o/p

(3) Effectiveness (each instruction must be very basic)

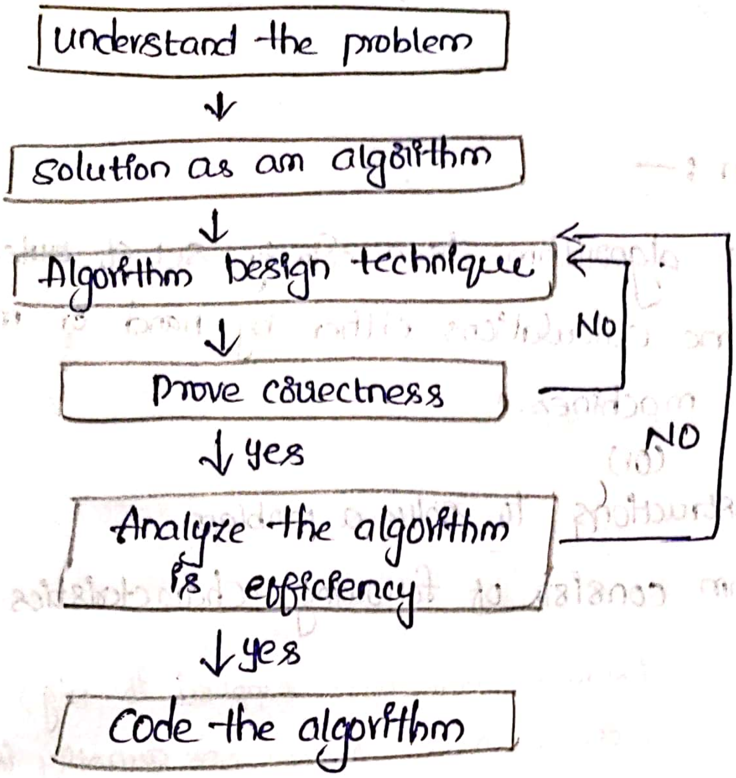
(4) Finiteness (finite no.of steps) - we can trace out the instructions of an alg for all the cases the alg must be terminate after a finite number of steps.

(5) Definiteness

↳ Each instruction is Clear & unambiguous.

Criteria for Algorithms...
Input : Zero or more inputs
output : Atleast one output
Finiteness : N number of steps
Defitness : Clear algorithm step
Effectiveness : A carried out step

⇒ Process of design & Analysis of Algorithm



Algorithm Design Techniques :

we can use different alg techniques

1. Divide & Conquer method
2. Greedy method
3. Dynamic programming
4. Back tracking
5. Branch & Bound ...

* Specifications of an algorithm :

There are various ways like which we can specify an algorithm

(1) Using Natural Language
(2) pseudo code
(3) flow chart
(4) Program (by using any programming language)

Natural Language

E.g:- Write an algorithm to perform addition of 2 no.s

Step 1:- Read a first number say 'a'

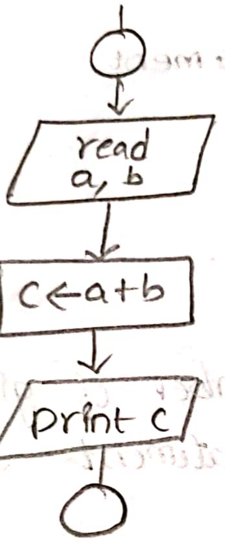
Step 2:- Read a Second number say 'b'

Step 3:- Add 2 no.s and store the resultant value in 'c'

Step 4:- Display the result.

Flow chart

Draw a flow chart for addition of 2 numbers



* **Pseudo code for expressing algorithms**

There are some general procesures for writing the Pseudo code

⇒ Comment lines begin with //

⇒ A block of statements represented by using { }

⇒ Example:- In if statement, for loop, while loop

```
if (a>0)
{   c = a+b;
    b = b+1;
}
```

⇒ An identifier begin with letter followed by (l+d).
e.g:- a, toy, sum1, a1102, hi.....

⇒ Assign the values to the variable by using assignment operator ':='

⇒ There are two boolean values True & false

⇒ Logical operator AND, OR, NOT

⇒ Relational operator <, <=, >, >=, =, != ....

⇒ Arithmetic operator +, −, %, /

⇒ The conditional statement if than or if than else written as follows

if (Condition) then (statement)

if (condition)
    (Statement)
else
    (Statement)

e.g:- find the largest number of given two no.s by using conditional statement

read a, b
if (a > b)
{ [printf("%d",a)]; write a is big
}
else
{ [printf("%d",&b)]; write b is big
}

⇒ Case statements

switch (condition)
{ Case (Condition)
    Statement − I
  Case (Condition)
    Statement − II
    ....
}

→ Loop statements

The general form of for loop is

for (variable = value1 to value N steps

{ }

e.g:- display 1 to 10 numbers by using for loop

```
for ( i = 1 to 10 )
do{
    write i
    i = i+1
  }
```

→ while Loop

The general form of while loop is

```
while ( condition ) do
{   stat-I
    state-II
      :
      N
}
```

e.g:- display 1 to 10 no.s by using while loop

```
~~read~~ i=1
while ( i ≤ 10)
{   write i
    i = i+1
}
```

28/12/22

10. Repeat until loop

General form is

```
repeat
{
   st-1
   st-2
    :
   st-n
} until (condition)
```

e.g:- Display 1 to 10 numbers

```
i=1
repeat
{ write(i);
  i=i+1;
} until (i ≤ 10);
```

11. Break statement : It is used to exit the loop.

12. The elements of an array are accessed by [ ]

e.g:- a=[1,2,3,4,5]

1d array - a[10]

2d array - a[2][3]

13. Delimiter → ; is used at the end of each statement

14. function / procedure

e.g:- algorithmname (parameter list)   ← prod name

```
{
    Statement 1 ;
    ⋮
    Statement n ;
}
```

Body of proced

15. Compound datatype can be performed with records

syntax : Name = record

```
{
    datatype1 - data1 ;
    datatype2 - data2 ;
    ⋮
}
```

e.g:- Write Algorithm for sum of n numbers

```
Algorithm Sum(n)
{   total:=0;
    for i:=1 to n do
    total := total+i;
    i := i+1;
}
```

e.g:- Write Algorithm for matrix multiplication Algorithm

```
Algorithm multiplication(A,B,n)
{   for i:=1 to n do
        for j:=1 to n do
            c[i,j]:=0;
            for k:=1 to n do
                c[i,j]:= c[i,j]+A[i,k]*B[k,j];
}
```

Performance Analysis :-

The performance Analysis or the analysis of an algorithm refer to the task of efficiency of an Algorithm i.e How much computing time and storage that an algorithm require to run or execute.

. The Analysis of an algorithm helps to comparing the two algorithms

. The Analysis can be done in two ways

1. Time complexity 2. Space complexity

Time Complexity :-

The time complexity of an Alg is the amount of compile time it needs to run or execute.

we can measure the time complexity of an Algorithm in 2 Approaches.

1. Compile time
2. Execution time

29/12/22.

e.g:- Time Complexity $T(P) = C + T_P(n)$.

e.g:- Write an algorithm for sum of n numbers

~~[for(i=0; k=n; i++)]~~

```
Algorithm Sum(n)
{     total := 0;
      for i := 1 to n do
      total := total + i;
      i := i+1;
}
```

Algorithm sum	0
{	
sum = 0	1
for i=1 to n do	n+1
{	
sum = sum+i;	n
i++;	n
}	
return sum;	1
}	
	3n + 3

(or) $\Rightarrow O(n)$

Alg sum(a,n)	0
{	
sum = 0;	1
for i=1 to n do	n+1
sum = sum + a[i];	n
return sum	1
}	
	2n+3

$\Rightarrow O(n)$

Space Complexity :-

The space complexity of an algorithm is the amount of memory it needs to run or completion. The space needed by an algorithm has the following components

i) Instruction Space

ii) Data Space

iii) Environment Stack Space.

i) Instruction Space :-

It is space needed to store the compiled version of the program instruction. The amount of I.S that is needed depends on following factors

a) The compiler used to compile the program into machine code

b) The compiler options in effect at the time of compilation.

c) The target computer i.e computer that is in which the algorithm run.

Note :-

One compiler may produce less code as compared to another compiler when executing the same program.

ii) Data Space :-

It is needed to store all constants & variable values. The D.S has two components

(a) Space needed by constants

(b) Space needed by dynamically allocated objects.

iii) Environmental stack space :-

It is used during execution of functions at each time the fun^n is involved in the following data are saved as the environmental stack. The data is

(a) The return address

(b) The value of local variables

(c) Value of formal parameters in the functions.

→ It is mainly used in recursive functions.

Scanned with CamScanner

→ The space requirement of any program P can be written as

Space complexity $S(P) = C + S_P$ (Instance characteristics)

This eqⁿ gives the total space needed by a program that is divided into 2 parts

(1) Fixed space requirement (c)

(2) Instance characteristics (S_P)

example :—

3/1/23

Time Complexity

1. For comment lines and declaration the step count value is zero
2. The return values & assignment statements the step count value is 1
3. Ignore the lower level exponential values compared with high level exponential values are present.

e.g :— $3n^4 + 10n^3 + 20n^2 + 100n + 2000$

$T(P) = O(n^4)$

	Steps for execution	frequency	total no. of steps
Algorithm Sum (a, b, c, m, n)	0	-	0
{	0	-	0
for i = 1 to m do	1	m+1	m+1
for j = 1 to n do	1	m(n+1)	mn+m
c[i][j] = a[i,j] + b[i,j]	1	mn	mn
}	0	-	-

$2m + 2mn + 1$

$\Rightarrow O(mn)$

e.g :—

```
A()
{ int i;              → 0
  for i ← 1 to n do   → n+1
     printf("WIT");   → n
}
```

$2n+1$

$\Rightarrow O(n)$

e.g.- A( )

{ int i,j,k,n; → 0

for(i=1; i<n; i++) → $n+1$

{ for(j=1; j≤i; j++) → $n(n+1) = n^2+n$

{ for(k=1; k≤100; k++) → $n \times n \times 100 = \frac{n^2}{100}$

pf("UVIT"); → 100

}

} }

$101n^2+2n+101$

$O(n^2)$

i=1	i=2	i=n
j=1	j=2	j=n
k=1*100	k=2*100	k=n*100

$\Rightarrow 1*100+2*100+ \cdots + n\times 100$

$\Rightarrow 100(1+2+\cdots+n) = \frac{100(n(n+1))}{2} = 50n^2+50$

$\Rightarrow O(n^2)$

e.g.- A( )

{ i=1; → 1

for i=1 to √n do → $\sqrt{n}+1$

pf("UVIT"); → $\sqrt{n}$

}

$2\sqrt{n}+2 \Rightarrow O(n^{1/2})$

A( )

{ i=1 → 1

for(i=1; i²≤n; i++) → $\sqrt{n}+1$

pf("UVIT"); → $\sqrt{n}$

}

$2\sqrt{n}+2 \rightarrow O(\sqrt{n})$

e.g.- A()

{ int i,j,k,n; → 1 = 1

for(i=1; i<n; i++) → $n+1$ = $n+1$

for(j=1; j≤i²; j++) → n^2 = n^2

for(k=1; k=n/2; k++) → $n\times n^2\times(\frac{n}{2}+1) = \frac{n^4}{2}+n^3$

pf("UVIT"); → $n\times n^2\times \frac{n}{2} = \frac{n^4}{2}$

}

$\Rightarrow O(n^4)$

Scanned with CamScanner

$i=1$, $j=1$, $k=1 \times n/2$ | $i=2$, $j=4$, $k=4 \times n/2$ | ...... $i=n$, $j=n^2$, ... $k=\frac{n^2}{2} \times \frac{n}{2} = \frac{n^3}{2}$

$(1+\frac{n}{2})(\ldots n+n)(\ldots$

$(1 * \frac{n}{2})(2 * \frac{n}{2})(\ldots (n^2 \times \frac{n}{2}) \Rightarrow \frac{n}{2} \times \frac{n(n+1)(2n+1)}{12}$

Time complexity for $\Rightarrow O(n^4)$

Recursive Algorithms

e.g:- A( )

```
{ if (condition)
    return ( A(n/2) + A(n/2) );
}
```

$T(n) = C + 2T(n/2)$

Some C=constant time For A(n) ↓ T(n) be the time Complexity

e.g:- A(n)

```
{ if (n>1)
  return A(n-1)
}
```

$T(n) = c + T(n-1)$
$= 1 + T(n-1)$; if $n>1$
$= 1$; if $n=1$

$T(n) = 1 + T(n-1)$ — ①
$T(n-1) = 1 + T(n-2)$ — ②
$T(n-2) = 1 + T(n-3)$ — ③

$\Rightarrow T(n) = 1 + T(n-1)$
$= 1 + T(n-2) + 1 = 2 + 1 + T(n-3)$
$= 3 + T(n-3) = 3 + 1 + T(n-4) = 4 + T(n-4)$
⋮
$k + T(n-k)$

Let $n-k=1 \Rightarrow n-1=k$

$\Rightarrow T(n) = n-1 + T(\cancel{n}-\cancel{n}+1) = n-\cancel{1}+\cancel{1} = n$

$\Rightarrow O(n)$

e.g:-

$T(n) = n + T(n-1)$, if $n>1$
$= 1$ if $n=1$

Find the time complexity by using back substitution method

Ans:- $T(n) = n + T(n-1)$; if $n > 1$:

$T(n-1) = n + T(n-2)$

$T(n-2) = n + T(n-3)$

$T(n-3) = n + T(n-4)$

$n - k = 1$

$k = n-1$

$T(n) = n + T(n-k)$

$= n + T(n - n + 1) = n + 1 \Rightarrow O(n)$

$T(n-k) = n - k + T(n-k+1)$

$T(n) = n + T(n-1)$ if $n > 1$

$T(n-1) = (n-1) + T(n-2)$ $(k-1) = 1$

$T(n-2) = (n-2) + T(n-3)$ $k = 2$

$T(n-3) = (n-3) + T(n-4)$

$T(n-k) = k + T(k-1)$

$= 2 + T(1) = 2 + 1 = 3$

$T(n) = n + T(n-1)$

$n - k = 1$

$k = n-1$

$T(n) = n + T(n-1)$

$= n + (n-1) + T(n-2)$

$= n + (n-1) + (n-2) + T(n-3)$

$\vdots$

$n + (n-1) + (n-2) + \dots + (n-k) + T(n-k)$

$= n + (n-1) + (n-2) + \dots + 1 + T(1)$

$= n + (n-1) + (n-2) + \dots + 1$

$= 1 + (1 + 2 + \dots + n) = 1 + \frac{n(n+1)}{2}$

$\Rightarrow O(n^2)$ //

Q. Calculate time complexity for the given recurrence relation $T(n) = 2T(n/2) + n$; otherwise

$= 1$; if $n = 1$

Soln:- Given $T(n) = 2T(n/2) + n$

$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2/2}\right) + n/2 = 2T(n) + n/2$

$T\left(\frac{3n}{2}\right) = 2T(3n) + \frac{3n}{2}$

$T(5n/2) = 2T(5n) + 5n/2$

$T(n) = 2T(n/2) + n$

$T(\frac{n}{2}) = 2T(\frac{n}{4}) + \frac{n}{2}$ — ①

$T(\frac{n}{4}) = 2T(\frac{n}{8}) + \frac{n}{4}$ — ②

$T(\frac{n}{8}) = 2T(\frac{n}{16}) + \frac{n}{16}$ — ③

$\Rightarrow T(n) = 2T(\frac{n}{2}) + n$

$= 2[2T(\frac{n}{4}) + \frac{n}{2}] + n$

$= 4T(\frac{n}{4}) + \frac{n}{\cancel{2}} + n$

$= 4[2T(\frac{n}{8}) + \frac{n}{4}] + \frac{n}{\cancel{2}} + n$

$= 8T(\frac{n}{8}) + n + n + n$

$= 8T(\frac{n}{8}) + 3n = 2^3 T(\frac{n}{2^3}) + 3n$

$\vdots$

$= 2^k T(\frac{n}{2^k}) + kn$

Let $\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log_2(n)$

$= 2^k T(\frac{\cancel{2^k}}{\cancel{2^k}}) + \log_2(n)\, n$

$= n\cancel{2^k}(1) + \log_2(n)\, n$

$= n \log_2(n) + \cancel{2^k}\, n \Rightarrow (n+1)(1 + \log_2(n))$

$\Rightarrow \underline{O(n \log_2 n)}$

* Mas

$T(n) =$

$a \geq 1,$

1) If a

2) If a

a)

b)

c)

3) If

exa

$T($

(Q) A

* Master Theorem

4/1/23

$T(n) = aT(n/b) + \Theta(n^k \log^p n)$

$a \geq 1$, $b > 1$, $k \geq 0$ and p is Real number

1) If $a > b^k$ then $T(n) = \Theta(n^{\log_b a})$
2) If $a = b^k$
   a) If $p > -1$ then
      $T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$
   b) If $p = -1$ then
      $T(n) = \Theta(n^{\log_b a} \log \log n)$
   c) If $p < -1$ then
      $T(n) = \Theta(n^{\log_b a})$
3) If $a < b^k$ then
   a) If $p \geq 0$ then
      $T(n) = \Theta(n^k \log^p n)$
   b) If $p < 0$ then
      $T(n) = \Theta(n^k)$

$\log n = \log_2 4$
$k = \log$
$\log_2(n \ldots)$
$\log_2(n \ldots)$

example :- Find out time complexity by using master theorem for the given recurrance relation.

$T(n) = 3T(n/2) + n^2$

$a = 3$, $b = 2$, $k = 2$, $p = 0$

$a = 3$, $b^k = 2^2 = 4$

i.e $a < b^k$ then → ③

(a) here $p = 0$, then i.e case ③ ⓐ

$T(n) = \Theta(n^k \log^p n)$
$= \Theta(n^2 \log^0 n) = \Theta(n^2)$.

(Q) Find out T.C by using Master theorem for the given recurrance relation

$T(n) = 4T(n/2) + n^2$

Sol:- here $a = 4$, $b = 2$, $k = 2$, $p = 0$

$a = b^k$ i.e ②

(a) if $p > -1$ then $p > -1$ hence $T(n) = \Theta(n^{\log_2 4} \log n)$
$= \Theta(n^2 \log n)$

1) $T(n) = T(n/2) + n^2$

$a=1,\ b=2,\ k=2,\ p=0$

$a=1,\ b^k = 2^2 = 4$

i.e $a < b^k$ — 3 (a)

(a) If $p \geq 0$ then

$T(n) = \Theta(n^k \log^p n)$

$= \Theta(n^2 \log^0 n) = \Theta(n^2)$

(Q) $T(n) = 2^n T(n/2) + n^2$

$[a = 2^n,\ b=2,\ k=2,\ p=0]$

here $a = 2^n$ it is not a constant so the master theorem is not applicable.

(Q) $T(n) = 16T(n/4) + n$

$a=16,\ b=4,\ k=1,\ p=0$

$a=16,\ b^k = 4^1 = 4$

here $a > b^k$ then (1)

$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_4 16})$

$= \Theta(n^2)$.

(Q) $T(n) = 2T(n/2) + n\log n$

$a=2,\ b=2,\ k=1,\ p=1$

here $a=2,\ b^k = 2^1 = 2$

here $a = b^k$

and $p=1 \rightarrow$ (2)(a)

$T(n) = \Theta(n^{\log_b a} \log^{p+1} n) = \Theta(n^{\log_2 2} \log^2 n)$

$= \Theta(n \log^2 n)$.

(Q) $T(n) = 0.5T(n/2) + 1/n$

$= 0.5T(n/2) + n^{-1}$

$a = 0.5,\ b=2,\ k=-1,\ p=0$

$a = 0.5,\ b^k = 2^{-1} = 1/2 = 0.5$

not applicable

(here $a = b^k$ & $p=0 \rightarrow$ (2)(a))

$T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$

$= \Theta(n^{\log_2 0.5} \log n)$

$= \Theta(\frac{1}{n} \log n)$

(Q) $T(n) = 64T(n/8) - n^2 \log n$

$a=64,\ b=8,\ k=2,\ P=1$ → Not applicable

$[a=64,\ b^k = 8^2 = 64$

$a = b^k$ & $P=1$ ② ⓐ

$T(n) = \Theta(n^{\log_b a} \log^{P+1} n) = \Theta(n^{\log_8 64} \log^2 n)$

$= \Theta(n^2 \log^2 n)]$

(Q) $T(n) = 7T(n/3) + n^2$

$a=7,\ b=3,\ k=2,\ P=0$

$a=7,\ b^k = 3^2 = 6$ here $a > b^k$ & $P=0$ ③ ⓐ

$T(n) = \Theta(n^k \log^P n) = \Theta(n^2 \log^0 n) = \Theta(n^2)$ //

(Q) $T(n) = 2T(n/2) + n/\log n = 2T(\frac{n}{2}) + n \log^{-1} n$

$a=2,\ b=2,\ k=1,\ P=-1$

$b^k = 2^1 = 2$ here $a = b^k,\ P=-1$

$T(n) = \Theta(n^{\log_b a} \log\log n)$

$= \Theta(n^{\log_2 2} \log\log n)$

$= \Theta(n \log\log n)$ //

(Q) $T(n) = 3T(n/3) + \sqrt{n}$

$a=3,\ b=3,\ k=1/2,\ P=0$

$b^k = 3^{1/2}$

$a > b^k$ & $P=0 \Rightarrow T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_3 3})$ ①

$= \Theta(n)$ //

5/1/23

[* Asymtotic Notations]

Space Complexity

e.g.- Algorithm Sum (x, y, z)

{ return ~~[illegible]~~ x+y*z;

}

$S(P) = c + S_P$

x → 1
y → 1
z → 1

3

$C = 3,\ S_P = 0$

$S(P) = 3 + 0 = 3 \Rightarrow O(1)$

e.g:- Algorithm Sum(a, n) → a=1, n=1

```
{
    int total = 0;                  → 1
    for i=1 to n do
        total = total + a[i]        → Sp = a → n
}
```

$S(P) = C + S_P$ here $C = 3$, $S_P = n$

$= 3 + n$

$\hookrightarrow O(n)$

$i = 1$
$n = 1$
$total = 1$ } $C = 3$

$a[n] \rightarrow S_P = n$

e.g:- Algorithm sum(x, n) → n

```
{
    if (n >= 0) then
        return 0;
    else
        return sum(x, n-1, x[n]);   → 3
}
```

$S(P) = C + S_P$

$= 3 + n \Rightarrow O(n)$

* Asymptotic Notations

Asymptotic Analysis of Alg means comparing the relative performance of an Algorithm.

Some of the Asymptotic Notations are :

(1) Big-O — (worst case, upper bound)

(2) Big-Ω — (best case, lower bound)

(3) Big-Θ — (Average Case) (Both using upper & lower bound)

Mainly used (1)–(3)

(4) Little-ω

(5) Little-Θ

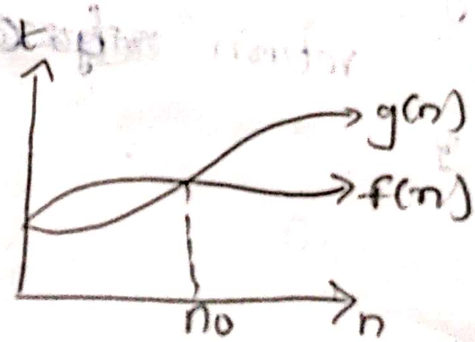
7/1/23.

(1) Big-O Notation

$f(n) \leq c \cdot g(n)$

$f(n) = g(n)$

$c > 0,\ n_0 \geq 1,\ n \geq n_0$



e.g:- $f(n) = 3n+2$ $\quad$ S.T $f(n) = O(g(n))$

$g(n) = n$

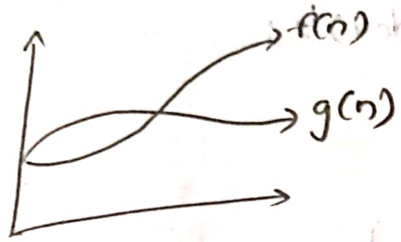
sol:- $f(n) \le c \cdot g(n)$

$\Rightarrow 3n+2 \le c \cdot n$ $\quad$ Consider $c=4$

$\Rightarrow 3n+2 \le 4n$ $\quad$ $n \ge 2$

$\Rightarrow 8 \le 8$

Big-Ω

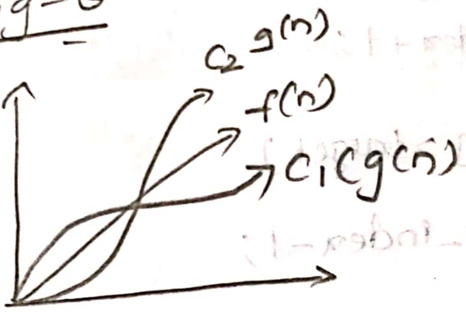


$f(n) \ge c \cdot g(n)$

$f(n) = g(n)$

$c > 0,\ n_0 \ge 1,\ n \ge n_0$

Big-Θ



$c_1 g(n) \le f(n) \le c_2 g(n)$

$n_0 \ge 1$ c, n Real no.s &

$n \ge n_0$

e.g:- $f(n) = 3n+2,\ g(n) = n$ $\quad$ S.T $f(n) = \Omega(g(n))$

sol:- $f(n) \ge c \cdot g(n)$ $\quad$ Consider $\boxed{c=1,\ n \ge 1}$ //.

$\Rightarrow 3n+2 \ge c \cdot n$

$\Rightarrow 3n+2 \ge 1 \cdot n$

$\Rightarrow 5 \ge 1$

e.g:- $f(n) = 3n+2,\ g(n) = n$ $\quad$ S.T $f(n) = \Theta(g(n))$

$c_1 g(n) \le f(n) \le c_2 g(n)$ $\quad$ $c_1 = 1$ $\quad$ $c_2 = 5$

$1 \cdot n \le 3n+2 \le 5 \cdot n$ $\quad$ $n \ge 1$

$1 \le 5 \le 5$

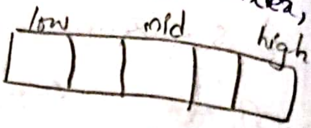
$\therefore$ $c_1 = 1$ & $c_2 = 5$

$n \ge 1$ //.

* Divide & Conquer Method

9/1/23

Binary Search Algorithm




```
Int binarySearch ( int arr[], int L-index, int r_index, int target)
{
    int m_index;
    while (l_index <= r_index)
    {
        m_index = (l_index + r_index)/2;
        if (arr[m_index] == target)
            return m_index;
        if (arr[m_index] < target)
            l_index = m_index + 1;
        else
            if (arr[m_index] > target)
                r_index = m_index - 1;
    }
    return 1;
}
```

Recurrence Relation $T(n) = \begin{cases} T(n/2)+1 & ; n \geq 1 \\ 1 & ; n = 1 \end{cases}$ to check key / Not

Sol^n:-

$T(n) = T(n/2)+1$

$T(n/2) = T(n/4)+1$ — ①

$T(n/4) = T(n/8)+1$ — ②

$T(n/4) = T(n/8)+1$ — ③

$T(n) = T(n/2)+1$

$= T(n/4)+1+1$

$= T(n/8)+3 = T(n/2^3)+3$

$= T(\frac{n}{2^1}) + T(\frac{n}{2^2}) + T(\frac{n}{2^3}) + \cdots + T(\frac{n}{2^k}) + k$

$= T(1) + \log_2^n = 1 + \log_2^n$

$\frac{n}{2^k} = 1$

$n = 2^k$

$\log n = k \log 2$

$\Rightarrow k = \log_2^n$

$\Rightarrow O(\log_2^n)$

Merge Sort

e.g.:-

Merge Sort

e.g:- | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

⇒ | 1 | 5 | 7 | 8 | 2 | 4 | 6 | 9 |
0 1 2 3 4 5 6 7

(1) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

| 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | here i<j , i++ = 2

(2) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | here i<j , j++ = 5

(3) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | i≮j , j++ = 6

(4) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | i < j i++ = 3

(5) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | i≮j j++ = 7

(6) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | i<j i++ = 4

(7) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 | i<j i++ = 4

Merge Sort

e.g:- | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

⇒ | 1 | 5 | 7 | 8 | 2 | 4 | 6 | 9 |
0 1 2 3 4 5 6 7

(1)
0 1 2 3 — | 1 | 5 | 7 | 8 |
4 5 6 7 — | 2 | 4 | 6 | 9 |
i ↑ j

| 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 |

here i<j , i++ = 2

(2)
0 1 2 3 — | 1 | 5 | 7 | 8 |
4 5 6 7 — | 2 | 4 | 6 | 9 |
↑ j

here i<j , j++ = 5

(3)
0 1 2 3 — | 1 | 5 | 7 | 8 |
4 5 6 7 — | 2 | 4 | 6 | 9 |
i ↑ j

i≮j , j++ = 6

(4) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

i < j i++ = 3

(5) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

i≮j j++ = 7

(6) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

i<j i++ = 4

(7) | 1 | 5 | 7 | 8 | | 2 | 4 | 6 | 9 |

i<j i++ = 4

18/01/23

2. The Greedy Method

UNIT-II

* Job Sequencing with deadlines

Consider a sequencing problem where there are 4 Jobs those are J1, J2, J3 & J4

	J1	J2	J3	J4
Profit	50	15	10	25
deadline	2	1	2	1

⇒

	J3	J2	J4	J1
profit	10	15	25	50
deadline	2	1	1	2

⇒ (J1, J4, J2, J3) (In descending order)
(2, 1, 1, 2)

⇒ J2, (J1, J3)
J4, (J1, J3)
(or)

⇒ Draw the Gantt chart based on the deadlines of the jobs
here the deadline is 2
Then the Gantt chart size is 2

50 10 = 60

J1	J3

0 1 2 ⊗

The combination of Jobs J1, J3 we can get the profit here 60

25 50 = 75

J4	J1

0 1 2 ✓

Here the profit is high. Hence we choose this Gantt chart.

15 50 = 65

J2	J1

0 1 2 ⊗

10 10 = 60

J3	J4

0 1 2 ⊗

15 25 = 35

J4	J3

0 1 2 ⊗

∴ We can get the max profit by selecting the sequence of jobs by completing is
(J4, J1)
max profit = 75.

② consider a sequencing ... having the profit of ... deadlines d(2, 3, 1, 4)

Sol:- The problem here

	J1	J2	J3	J4
P	10	30	60	40
d	2	3	1	3

The descending order
(J3, J4, J2, J1)
(1, 3, 3, 2)

⇒ Here Gantt chart

J3	J1	J...

0 1 2

⇒

J3	J2

0 1

The

19/1/23

③ find out ... no. of Jobs a... given below

	J1
Profit	2
weight	3

Sol:- The d...

J5	J2
4	3

⇒ Here

0

UNIT-II

② Consider a sequencing problem where there are 4 jobs having the profit of P(10, 30, 60, 40) and the corresponding deadlines d(2, 3, 1, 4) find out the max profit by using Greedy method.

Sol:- The problem here is

where n = 4 (no. of Jobs)

	J1	J2	J3	J4
P	10	30	60	40
d	2	3	1	3

The descending order of Jobs based on profit is

(J3, J4, J2, J1)

(1, 3, 3, 2)

⇒ Here Gantt chart size is 4

J3	J1	~~J2~~ J4	~~J4~~
0–1	1–2	2–3	3–4

⇒ 60 + 10 + 40 = 110

⇒

J3	J2	J4
0–1	1–2	2–3

⇒ 60 + 30 + 40 = 130

The max. profit = 130 by selecting the Jobs (J3, J2, J4)

19/1/23

③ find out the max profit by using Greedy method. The no. of Jobs are 4 & corresponding profit & weights are given below

	J1	J2	J3	J4	J5
Profit	2	4	3	1	10
weight	3	3	3	4	4

Sol:- The descending order of Jobs

J5	J2	J3	J1	J4
4	3	3	2	1

⇒ Here the Gantt Chart size is 4

J1	J3	J2	J5
0–1	1–2	2–3	3–4

⇒ 19.

The max. profit = 19

④ Find out the maximum profit by using Greedy method. The no. of jobs corresponding weights & profits given below

	J1	J2	J3	J4	J5	J6	J7	J8	J9
profit	15	20	30	18	18	10	23	16	25
wet	7	2	5	3	4	5	2	7	3

Sol:- The descending order of Jobs based on profit is

(J3, J9, J7, J2, J4, J5, J8, J1, J6)
5 3 2 2 3 4 7 7 5

The Gantt chart size is 7

J2	J7	J9	J5	J3	J1	J8

0 1 2 3 4 5 6 7

⇒ 20 + 23 + 25 + 18 + 30 + 15 + 16

⇒ 147

20 + 23 + 25 = 68, + 30 + 18 = 116, + 15 + 16 = 147

The max profit is ⇒ 147
by selecting the jobs
(J2, J7, J9, J5, J3, J1, J8).

⑤ Find out the max profit by using Greedy methods. All the jobs are given below along with deadline & profits

Job	J1	J2	J3	J4	J5
D	2	1	3	2	1
P	60	100	20	40	20

Sol:- The descending order of Jobs based on profit is

1 2 2 3 1
(J2, J1, J4, J3, J5)
100 60 40 20 20

The Gantt chart size is 3

J2	J1	J3

0 1 2 3

= 100 + 60 + 20 = 180

max profit is 180 by taking the jobs in sequence of (J2, J1, J3)

⑥

Jobs	J1
D	2
P	100

Sol:- the desce... is (J1, J3 ...
2 2

The Gantt ...

J3

0

The max...
sequence

23/1/23 Knap...

10. Find a ma... M = 20, n... below

Sol:- Step1: G...
kn...
So, G...
item 3. Th...
Now, the ...
we can pla...
of the ...

⑥

Jobs	J1	J2	J3	J4	J5
D	2	1	2	1	3
P	100	19	27	25	15

Sol^n:- The descending order of Jobs based on profit is (J1, J3, J4, J2, J5)
2 2 1 1 3

The Gantt Chart size is 3

J3	J1	J5

0 1 2 3

$= 27 + 100 + 15$

$= 142$

127
15
142

The max profit is 142 by taking the Jobs in sequence of (J3, J1, J5)

23/1/23

Knapsack problem:-

Q. Find a max profit by using knapsack problem where M = 20, n = 3 the corresponding wts & profits are given below.

Where M = the wt of knapsack bag
n = no. of items

	item 1	item 2	item 3
Profit	25	24	15
Wt	18	15	10

Sol:- Step1: Greedy about wt

knapsack bag wt M = 20

so, Greedy about wt, 1st we can store or place item 3. The item size is 10

W	P
10	15

Now, the remaining bag size is 20−10 = 10

but the capacity of the knapsack is only 10

we can place item 2. The item size is 15

W	P
10	15
10	$\cancel{24}^{8}\left(\frac{\cancel{10}^{2}}{\cancel{15}_{3}}\right) = 16$

Total : 31

Step 2: Greedy about profit

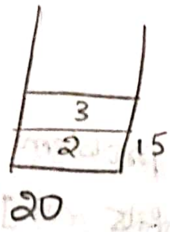
P	W
25	18
$\cancel{24}^{8}\left(\frac{2}{\cancel{15}_{5}}\right)$	2

$20 - 18 = 2$

$25 + \frac{16}{5} = \frac{125+16}{5} = \frac{141}{5} = 28.33$

Step 3: $\frac{\text{Profit}}{\text{wt}}$

	item 1	item 2	item 3
P/W	$25/18 = 1.38 \approx 1.4$	$24/15 = 1.6$	$15/10 = 1.5$



W	P
15	24
5	$\cancel{15}^{3} \times \left(\frac{5}{\cancel{10}_{2}}\right)$
	31.5

$20 - 15 = 5$

max profit = 31.5

2.) Find out the max. profit by using fractional knapsack problem or greedy problem where $M = 15$, $n = 7$. the profit & wt of objects are given below

Object	1	2	3	4	5	6	7
Profit (P)	10	5	15	7	6	18	3
wt (w)	2	3	5	7	1	4	1

Sol?—

P/W	5	1.6	3	1	6	4.5	3

Given data

$M = 15$

$n = 7$

Calculate the p/w ratio

W	P
1	6

$15 - 1 = 14$

W	P
1	6
2	10
4	18
5	15
1	3
3	$5 \times \frac{2}{3}$
	55.

The max. ...

the sequence ...

(3) Find a m...

Problem w...

profit & weig...

Sol?–

Obj	
Profit	
wt	
P/W	

	W
obj 2	4
obj 4	3
obj 3	5
obj 5	3

max prof...

The m...

In ...

w	p	
1	6	15-1 = 14
2	10	14-2 = 12
4	18	12-4 = 8
5	15	8-5 = 3
1	3	3-1 = 2
3	$5 \times \frac{2}{3} = \frac{10}{3} = 3.33$	

55.33

The max profit is 55.33 by arranging the items in the sequence of 5, 1, 6, 3, 7, 2

(3) Find a maximum profit by using Greedy knapsack problem where M=15, n=5 and the each & every object profit & weight is given below

Sol:-

obj	1	2	3	4	5
Profit	2	28	25	18	9
wt	1	4	5	3	3
P/w	2	7	5	6	3

	w	p	
obj2	4	28	15-4 = 11
obj4	3	18	11-3 = 8
obj3	5	25	8-5 = 3
obj5	3	9	3-3 = 0

max profit: 80

28
+18
46
+25
71
9
80

The max profit is 80 by arranging the items in the sequence of 2, 4, 3, 5

(4) Find out a max. profit by using Greedy Knapsack Problem where $M=60, n=5$. The Profit & wts are given below.

Sol^n:- Given data

Item	1	2	3	4	5
w	5	10	15	22	25
P	30	40	45	77	90
P/w	6	4	3	3.5	3.7

	P	w
obj 1	30	5
obj 2	40	10
obj 5	90	25
obj 4	$\cancel{20 \times 77}$ / $\cancel{22}$ 7 → 70	20

$60-5=55$

$55-10=45$

$45-25=20$

max profit = 230

The max profit is 230 by arranging the objects in sequence of 1, 2, 5, 4

(5) Find out the max. profit by using Greedy Knapsack problem where

$M=15$ $n=7$

Sol^n:-

obj	1	2	3	4	5	6	7
P	5	10	15	7	8	9	4
w	1	3	5	4	1	3	2
P/w	5	3.3	3	1.75	8	3	2

	P	w	
obj 5	8	1	$15-1=14$
obj 1	5	1	$14-1=13$
obj 2	10	3	$13-3=10$
obj 3	15	5	$10-5=5$
obj 6	9	3	$5-3=2$
obj 7	4	2	$2-2=0$
	$\boxed{51}$		

The max profit is 51 by arranging the objects in sequence of 5, 1, 2, 3, 6, 7 //

(6) Find the max. p... $M=60, n=4$

Sol^n:-

obj	1	2	3
P	280	100	120
w	40	10	20
P/w	7	10	6

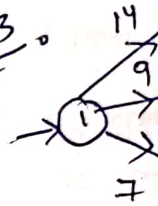
25/1/23

\# Single Sour...

ex:- Fi...

Shortest pa...

27/1/23



Ans:-

Source
1
1, 2
1, 2, 3
1, 2, 3, 6
1, 2, 3, 6, 4
1, 2, 3, 6, 4...

(b) Find

$M = 60, n = 4$

sol:-

obj	1	2	3	4
P	280	100	120	120
W	40	10	20	24
P/W	7	10	6	5

using fractional knapsack problem

	P	W
obj 2	100	10
obj 1	280	40
obj 3	60	10
	440	

60 − 10 = 50

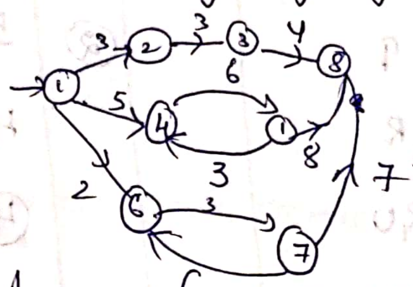
50 − 40 = 10

The max profit is 440 by arranging in sequence of obj 2, 1, 3 //.

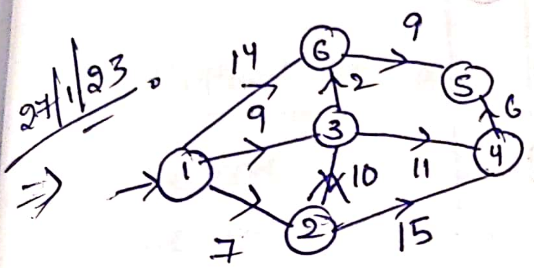
25/1/23

* Single Source Shortest path.

ex:- Find out the shortest path by using single source shortest path technique



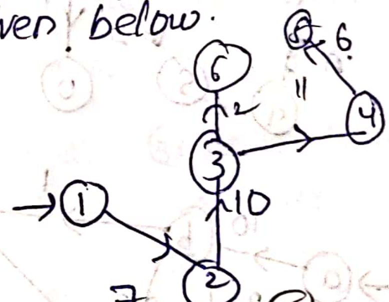
27/1/23

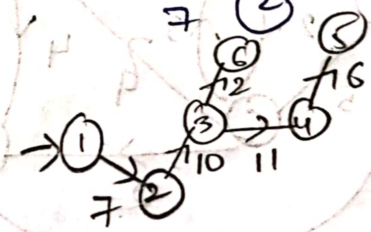


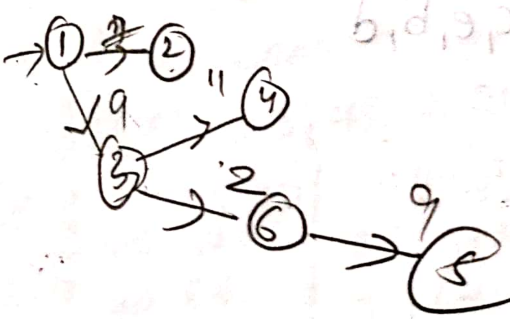
Find out a shortest Path by using single source shortest path (Greedy alg) for the graph given below.

Ans:-

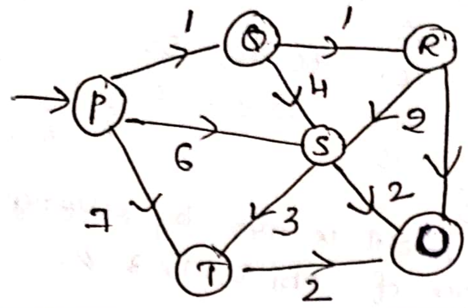
Source	2	3	4	5	6
1	7	9	∞	∞	14
1, 2		9	22	∞	14
			20	∞	11
1, 2, 3					
1, 2, 3, 6			20	20	
1, 2, 3, 6, 4					
1, 2, 3, 6, 4, 5					



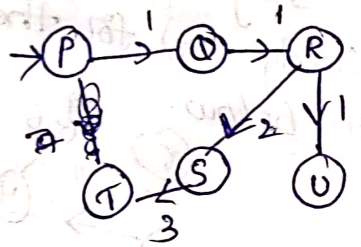




(3)

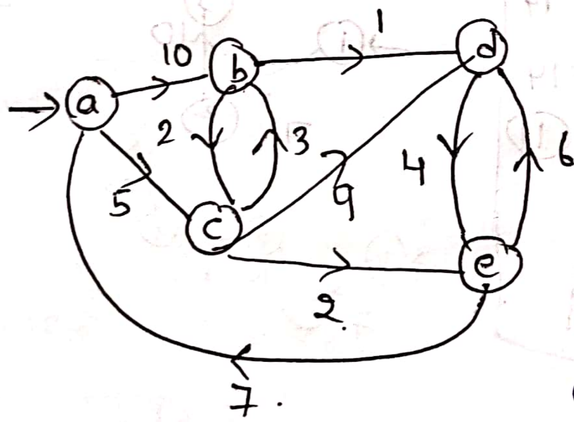


Source	~~P~~	Q	R	S	T	U
→ P		1	∞	6	7	∞
P, Q			2	5	7	∞
P, Q, R				4	7	3
P, Q, R, U				4	7	
P, Q, R, U, S, T					7	

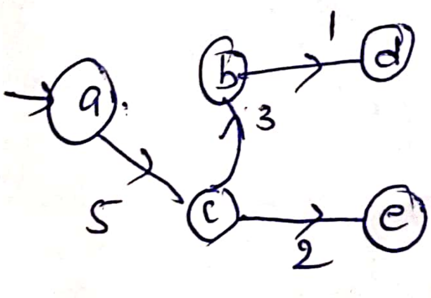


Total path:- ⇒ 8

(4)



Source	b	c	d	e
a	10	5	∞	∞
a, c	8		14	7
a, c, e	8		9	
a, c, e, b			9	
a, c, e, b, d				



(5)

→(2

Source
1
4,2
1,2,5
1,2,5,4,
1,2,5,4,

28/1/23. Opt

Q.) Find the
these are
5,18,26,4,

Ans:- Step1:-

Step2:-
T0 3
T1 4
T2 5

Step3:- f

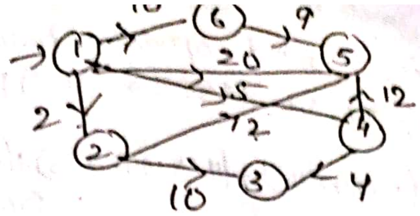
Q.) Find
of e
are

Ans:- 8

T0
T1
T2

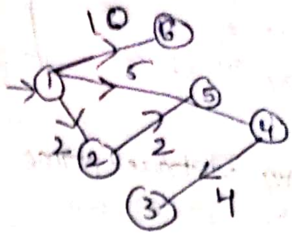
Avg
(or)
Mean

⑤



Ans :- 1, 2, 4, 3, 6, 5

Source	2	3	4	5	6
1	2	∞	5	20	10
1, 2		12	5	4	10
1, 2, 5		12	5		10
1, 2, 5, 4,		9			10
1, 2, 5, 4, 3, 6					10



21/1/23. Optimal Storage on tapes using Greedy Method

Q.) Find the optimal placement for 13 programs in on 3 tapes. these are T0, T1, T2 where the program lengths are 10, 5, 8, 32, 7, 5, 18, 26, 4, 3, 11, 10, 6. Find the optimal retrieval time.

Ans:- Step 1:- All the files are arranged in ascending order.

3, 4, 5, 5, 6, 7, 8, 10, 11, 12, 18, 26, 32

Step 2:-

T0	3	5	8	12	32	= 3 + (3+5) + (3+5+8) + (3+5+8+12) + (3+5+8+12+32) = 115
T1	4	6	10	18		= 4 + (4+6) + (4+6+10) + (4+6+10+18) = 72
T2	5	7	11	26		= 5 + (5+7) + (5+7+11) + (5+7+11+26) = 89

Step 3:- Avg retrieval time = $\frac{T0+T_1+T_2}{3}$ = 92

Q.) Find the optimal retrieval time where n = 10 and lengths of each files are 10, 20, 45, 70, 1, 3, 7, 54, 23, 67 and there are three tapes.

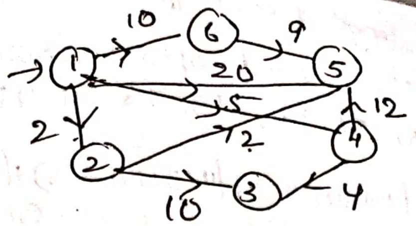
Ans:- Order ⇒ 1, 3, 7, 10, 20, 23, 45, 54, 67, 70

T0	1	10	45	70	1 + (1+10) + (1+10+45) + (1+10+45+70) = 194
T1	3	20	54		3 + (3+20) + (3+20+54) = 103
T2	7	23	67		7 + (7+23) + (7+23+67) = 134

Avg retrieval time (or) Mean = $\frac{T0+T1+T2}{3}$ = $\frac{194+103+134}{3}$ = 143

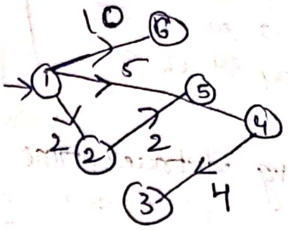
Scanned with CamScanner

(5)



Ans :- 1, 2, 4, 3, 6, 5

Source	2	3	4	5	6
1	2	∞	5	20	10
1, 2		12	5	4	10
1, 2, 5		12	5		10
1, 2, 5, 4,		9			10
1, 2, 5, 4, 3, 6					10



28/1/23. Optimal Storage on tapes using Greedy Method

Q.) Find the optimal placement for 13 programs in on 3 tapes those are T0, T1, T2 where the program lengths are 12, 5, 8, 32, 7, 5, 18, 26, 4, 3, 11, 10, 6. Find the optimal retrieval time.

Ans:- Step 1 :- All the files are arranged in ascending order.

3, 4, 5, 5, 6, 7, 8, 10, 11, 12, 18, 26, 32

Step 2 :-

T0	3	5	8	12	32	$= 3 + (3+5) + (3+5+8) + (3+5+8+12) + (3+5+8+12+32) = 115$
T1	4	6	10	18		$= 4 + (4+6) + (4+6+10) + (4+6+10+18) = 72$
T2	5	7	11	26		$= 5 + (5+7) + (5+7+11) + (5+7+11+26) = 89$

15
20
24
24
32
115

115
172
189
226
3

Step 3 :- Avg retrival time $= \dfrac{T0 + T_1 + T_2}{3} = 92$

Q.) Find the optimal retrieval time where $n = 10$ and lengths of each files are 10, 20, 45, 1, 3, 7, 54, 23, 67 and there are three tapes.

9
40
54
103

4
30
90
70
194

Ans :- Order ⇒ 1, 3, 7, 10, 20, 23, 45, 54, 67, 70

T0	1	10	45	70	$1 + (1+10) + (1+10+45) + (1+10+45+70) = 194$
T1	3	20	54		$3 + (3+20) + (3+20+54) = 103$
T2	7	23	67		$7 + (7+23) + (7+23+67) = 134$

21
46
67
134

Avg retrival time (or) $= \dfrac{T0 + T_1 + T_2}{3} = \dfrac{194 + 103 + 134}{3} = 143$

(Q.) n=13. There are 3 tapes & length of each file is 12, 34, 56, 7[illegible]
24, 11, 34, 56, 78, 91, 34, 91, 45. Find optimal retrieval time.
24, 11, 34, 56, 78, 91, 34, 91, 45, 56, 56, 73, 78, 91, 91

Sol:- 11, 12, 24, 34, 34, 34, 34, 45,

T0	11	34	45	73	91
T1	12	34	56	78	
T2	24	34	56	91	

$= 11+(11+34)+(11+34+45)+(11+34+45+73)+(11+34+45+[illegible])$
$= 563$
$= 12+(12+34)+(12+34+56)+(12+34+56+78) = 340$
$= 24+(24+34)+(24+34+56)+(24+34+56+91)$
$= 401$

Avg retrieval time $= \frac{T_0+T_1+T_2}{3}$

$= \frac{563+340+401}{3} = 434.6$

→ Optim[illegible]

Q. Find[illegible]
give[illegible]
Sol^n:-

→ Optimal Merge Patterns using Greedy method

24/1/23

Q. Find out total no. of merges or optimal merge for the given set of files 2,9,7,5,3,13

Sol:- Step 1:- Arrange them (files) in ascending order

2, 3, 5, 7, 9, 13 → merging

5, 5, 7, 9, 13 → check for order

merge

10, 7, 9, 13 → arrange in order

7, 9, 10, 13 — merge

16, 10, 13 → arrange in order

~~23~~ 10, 13, 16

23, 16 → order

16, 23 → merge

39

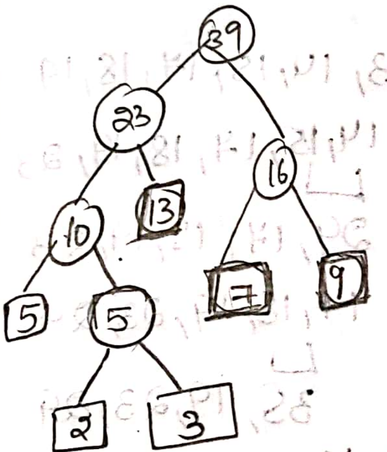
Total no. of merges = Sum of internal nodes

= 39+23+16+10+5

= 93

15
16
31
23
54

39
54
93



(Q.) The set of files are 2, 5, 7, 26, 11, 16, 21, 14, 38, 42

Sol:- ascending order ⇒ 2, 5, 7, 11, 16, 14, 21, 26, 38, 42

2, 5, 7, 11, 16, 14, 21, 26, 38, 42

7, 7, 11, 16, 14, 21, 26, 38, 42

14, 11, 16, 14, 21, 26, 38, 42

→ 11, 14, 16, 14, 21, 26, 38, 42

25, 16, 14, 21, 26, 38, 42

16, 14, 21, 25, 26, 38, 42

33, 21, 25, 26, 38, 42

→ 21, 25, 26, 33, 38, 42

46, 26, 33, 38, 42

→ 26, 33, 38, 42, 46

59, 38, 42, 46

Scanned with CamScanner

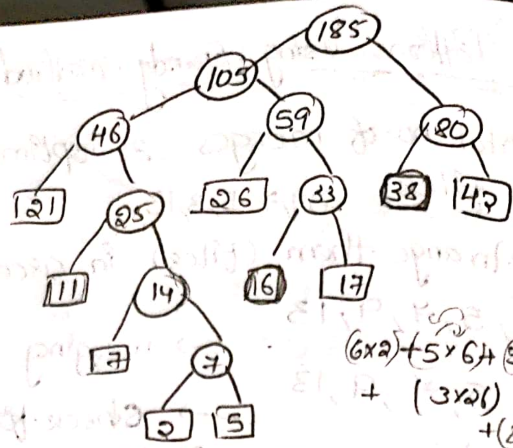
38, 42, 46, 59

80, 46, 59

→ 46, 59, 80

→ 80, 105

185



(6x2)+5x6+(5x7)+(4x11)+(6x...) + (3x21)+(4x16)+(4x17) +(2x38)+(2x42)

Total no. of merges = Sum of internal nodes

= 7+14+25+46+105+185+59+33+80 → distance

= 554 Formula Σdixi → leaf node

(Q) Construct optimal merge tree for the given files 8, 2, 9, 1, 12, 10, 18, 15, 14, 17

Sol:- Arrange the files in order each time & Perform merging.

1, 2, 8, 9, 10, 12, 14, 15, 17, 18

3, 8, 9, 10, 12, 14, 15, 17, 18

11, 9, 10, 12, 14, 15, 17, 18

→ 9, 10, 11, 12, 14, 15, 17, 18

19, 11, 12, 14, 15, 17, 18

→ 11, 12, 14, 15, 17, 18, 19

23

23, 14, 15, 17, 18, 19

→ 14, 15, 17, 18, 19, 23

29, 17, 18, 19, 23

→ 17, 18, 19, 23, 29

35, 19, 23, 29

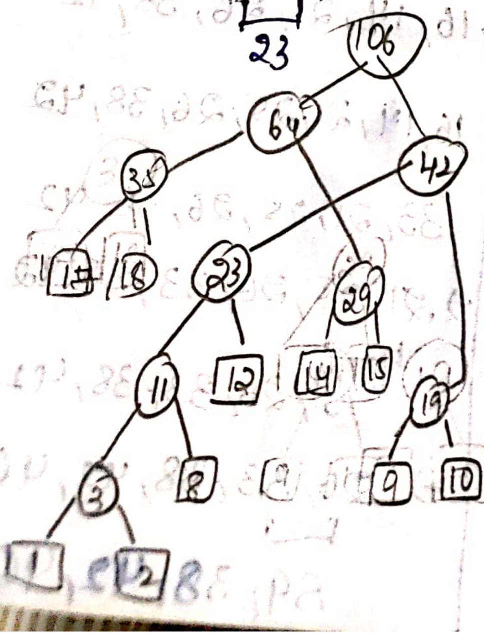
→ 19, 23, 29, 35

42, 29, 35

→ 29, 35, 42

→ 42, 64

106



Total no. of merges = Sum of internal nodes

= 106+64+42+35+23+29+11 +19+3

= 332

Finding a maximum ... method

(Q) Find a max & min ... size is 7 (n=7)

Sol:-

Max = 10

Min = 7

(Q) The array ... 9, 12, 6, 2.

Sol:-

Finding a maximum & minimum element by using divide & conquer method.

(Q) Find a max & min element using the given array. The array size is 7 (n=7) the elements: 7, 10, 13, 71, 32, 83, 2.

Sol:-

7	10	13	71	32	83	2

low 0 1 2 3 4 5 6 high

7	10	13	71

0 1 2 3

32	83	2

4 5 6

7	10

0 1

Max = 10
Min = 7

13	71

2 3

max = 71
min = 13

32	83

4 5

max = 83
min = 32

2

6

max = 2
min = 2

Max = 71
Min = 7

Max = 83
Min = 2

max = 83
min = 2

(Q) The array size n=8, the list of elements are 5, 7, 3, 4, 9, 12, 6, 2.

Sol:-

5	7	3	4	9	12	6	2

0 1 2 3 4 5 6 7

m=3

5	7	3	4

0 1 2 3

9	12	6	2

4 5 6 7

m=1

m=5

5	7

0 1

max = 7
Min = 5

3	4

2 3

max = 4
min = 3

9	12

4 5

max = 12
min = 9

6	2

6 7

max = 6
min = 2

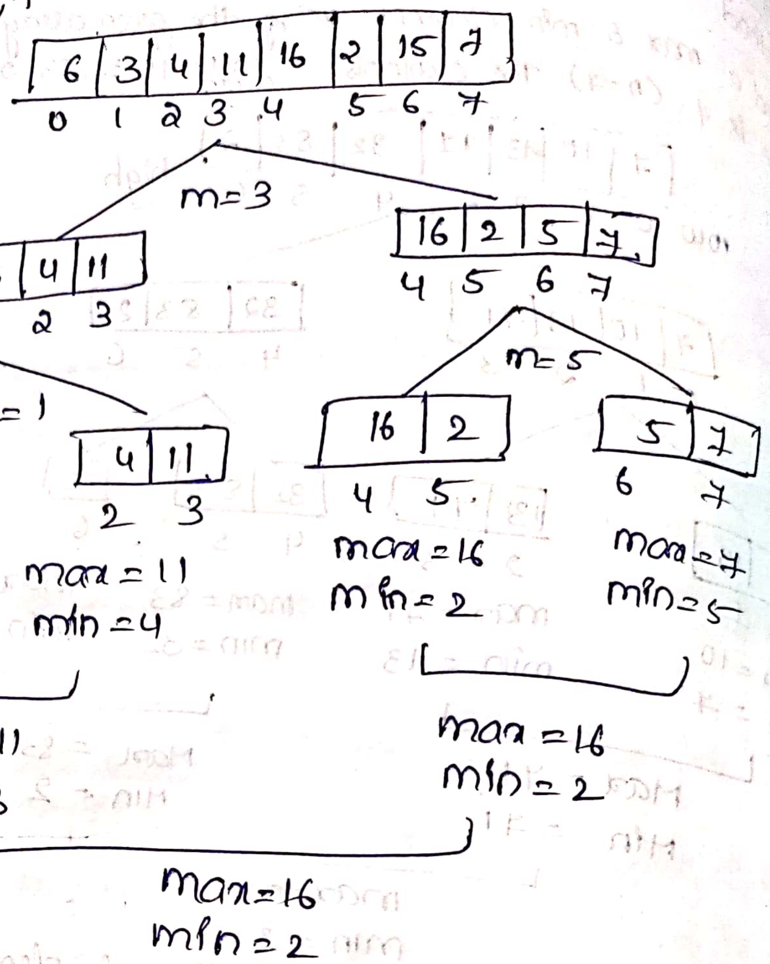
max = 7
min = 3

max = 12
min = 2

max = 12, min = 2

Q. Find a maximum & min element from the list 6, 3, 4, 11, 16, 2, 15, 7

Sol:-



Algorithm

```
Algorithm maxMin (low, high, max, min)
{
    if (low == high)
    {
        max = min = a[low];
    }
    else if (low == high-1)
    {
        if (a[low] > a[high])
        {
            max = a[low];
            min = a[high];
        }
        else
        {
            max = a[high];
        }   min = a[low];
```

else
{ mid
max
max
if (
if
}
}

Time com

Recurrence r

T(n) = 2T(

using Maste

a=2, b

$a > b^k$

$2 > 2^0$

use 1st

using

T(

T(

T(

else
{ mid = $\frac{low+high}{2}$;
maxmin (low, mid, max1, min1);
maxmin (mid+1, high, max, min);
if (max1 > max)
max = max1;
if (min1 < min)
min = min1;
}
}

Time complexity.

Recurrence relation is $T(n) = \begin{cases} 0 & 1 \\ 1 & 2 \\ 2T(n/2)+2 & n>2 \end{cases}$

$T(n) = 2T(n/2)+2$

using Master theorem: $T(n) = aT(n/b)+O(n^k \log^p n)$

$a=2, b=2, k=0, P=n^2.$

$a > b^k$

$2 > 2^0$

use 1st formula: $T(n) = O(n \log_b^a)$

$= O(n \log_2^2)$

$= O(n)$

using Backsubstitution method:

$T(n) = 2T(n/2)+2$

$T(n) = 2(2T(n/4)+2)+2$

$= 4(T(n/4)+4+2$

$T(n) = 4(2T(n/8)+2)+4+2$

$\vdots$

$= 2^k T(n/2^k) + 2^k + 2^{k-1} + \dots + 2^1$

$= 2^k (T(n/2^k)) + \frac{2\cdot(2^k-1)}{2-1}$ $\qquad \frac{a(r^n-1)}{r-1}$

$= 2^k(T(n/2^k)) + 2^{k+1} - 2$ $\qquad r=2, a=2, n=k$

$n/2^k = 2$

$2^k = n/2$

$\Rightarrow n/2 \left(T(2^{\cancel{k}}/\cancel{n}) \right) + n-2$

$\Rightarrow \frac{n}{2} + n - 2 \Rightarrow \frac{3n}{2} - 2 \Rightarrow \frac{3n-4}{2}$

$\Rightarrow O(n)$

Quick Sort:-

e.g:- 2, 9, 7, 5, 3, 13

pivot i, j

2	9	7	5	3	13
0	1	2	3	4	5

$i = 0$

$a[0] < a[j]$

$2 < 13$ ✓ i++

pivot i, j

2	9	7	5	3	13

$a[1] < a[5]$

$9 < 13$ ✓ i++

i, j

2	9	7	5	3	13
0	1	2	3	4	5

$7 < 13$

i++

i, j

2	9	7	5	3	13

$5 < 13$ ✓

i++

2 9 7 5 3 13

$3 < 13$ ✓

i++

$i == j$

Algorithm: {

newleft

mid =

while

while

Increa

Endw

while

Decre

Endw

If

Sw

Inc

End

Endw

If

Call

endi

If

Ca

E

}

Time comp

$T(n) = \{$

By using

T(n

T(n|

T(n|4

T(n|8

Algorithm : Quicksort (a, left, right)

{

newleft = left, newright = right

mid = a[left + right] / 2 // pivot

while newleft ≤ newright // partition

while (a[newleft] < [mid] and newleft < right) // while loop - 1

Increment newleft

Endwhile

while (a[mid] < a[newright] and newright > left) // while loop - 2

Decrement newright

Endwhile

If newleft ≤ newright then

Swap (a[newleft], a[newright])

Increment newleft and decrement newright

EndIf

Endwhile // partition ends

If left < newright then

Call Quicksort (a, left, newright) // Sort left subarray

endif

If newleft < right, then

Call Quicksort (a, newleft, right) // sort right subarray

EndIf

}

Time complexity

$$T(n) = \begin{cases} 1 \\ 2T(n/2) + n \end{cases}$$

By using backsubstitution method.

$T(n) = 2T(n/2) + n$

$T(n/2) = 2T((n/2)/2) + n/2 = 2T(n/4) + n/2$

$T(n/4) = 2T(n/8) + n/4$

$T(n/8) = 2T(n/16) + n/8$

$$T(n) = 2T(n/2) + n$$

$$= 2(2T(n/4) + n/2) + n$$

$$= 4(2T(n/8) + n/4) + n/2 + n$$

$$= 8(2T(n/16)) + n/8) + n/4 + n/2 + n$$

$$= 16T(n/16) + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n$$

$$\vdots$$

$$= 2^{k}T(n/2^{k}) + n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + \frac{n}{2^{k-1}}$$

$$= n\,T(1) + \cdots$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log n = k \log_2$$

$$k = \log_2^n$$

$$= O(n \log n)$$

$\therefore$ The time complexity for quick sort $O(n \log n)$

Average case time complexity $O(n \log n)$

Worst case time complexity $O(n^2)$

Merge sort algorithm

Algorithm Mergesort(left, right):

1. if (left < right)
2. mid = (left + right)/2
3. mergesort (left, mid)
4. Mergesort (middle, right)
5. mergesort (left, mid, right)
6. End if

Algorithm Merge (left, mid, right):

1. i = j = left, k = mid + 1
2. while $j \le mid$ and $i \le right$
3. if a[i] < a[k]

5. else
6. temp [i] =
7. End if
8. End whil
9. while $j \le$
10. Temp [i]
11. End whil

Time compl

0/1 kna

(Q) n=4,

Sol:-

Initially

i=0,

i=1,

i=2,

i=3,

$S^3 =$

$S_1^3 =$

S^3

$= 2(2T(n/4) + n/2) + n$

$= 4(2T(n/8) + n/4) + n/2 + n$

$= 8(2T(n/16)) + n/8) + n/4 + n/2 + n$

$= 16T(n/16) + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n$

$\vdots$

$= 2^k T(n/2^k) + n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + \frac{n}{2^{k-1}}$

$= n\,T(1) + \cdots$

$\frac{n}{2^k} = 1$

$n = 2^k$

$\log n = k \log_2$

$k = \log_2^n$

$= O(n\log n)$

$\therefore$ The time complexity for quick sort $O(n\log n)$

Average case time complexity $O(n\log n)$

Worst case time complexity $O(n^2)$

Merge sort algorithm

Algorithm Mergesort (left, right) :

1. if (left < right)
2. mid = (left + right)/2
3. mergesort (left, mid)
4. Mergesort (middle, right)
5. mergesort (left, mid, right)
6. End if

Algorithm Merge (left, mid, right) :

1. i = j = left , k = mid + 1
2. while j ≤ mid and i ≤ right
3. if a[i] < a[k]
4. Temp[i] = a[j] ;

5. else
6. temp[i] = a[k];
7. End if
8. End while
9. while $j \le mid$
10. Temp[i] = a[j];
11. End while

12. for i = left to k
13. a[i] = temp[j]
14. end for

Open book
Quick sort e.g. files
Pattern merging

Time complexity for Merge sort is $n\log(n)$

3. Dynamic Programming

UNIT – III

0/1 knapsack problem

(Q) $n=4,\ m=21,\ P=(2,5,8,1)$
$W=(10,15,6,9)$

Each & every profit & weights are given below

Sol:- $\boxed{\text{Formula :- } S^{i+1} = S^{i} \cup S_1^{i}}$

Initially $S^0 = (0,0)$

$i=0,\quad S^1 = S^0 \cup S_1^0$

$i=1,\quad S^2 = S^1 \cup S_1^1$

$i=2,\quad S^3 = S^2 \cup S_1^2$

$i=3,\quad S^4 = S^3 \cup S_1^3$

(2,10) $P_1\ W_1$
(5,15) $P_2\ W_2$

$S_1^0 = (0,0)\ (2,10)$

$S^1 = S^0 \cup S_1^0 = \{(0,0),(2,10)\}$

$S^2 = S^1 \cup S_1^1$

$S_1^1 = (5,15)$

$S_1^1 = \{(0,0),(2,10),(5,15)\}$

$= \{(5,15),(7,25)\}$ ✗

$P_1 < P_2 \quad W_1 > W_2$

$S^2 = \{(0,0),(2,10),(5,15),(7,25)\}$

$S^2 = \{(0,0),(2,10),(5,15)\}$

$S^3 = S^2 \cup S_1^2$

~~$S_1^2 = \{(0,0),(2,10),(5,15),(8,6)\}$~~

$= (7,25)\ \cancel{(8,6)} = (15,31)$

$S_1^2 = (8,6)\ (10,16)\ (13,21)$

$S^3 = \{(0,0),(2,10),(5,15),(8,6),(10,16),(13,21)\}$

$= \{(0,0),\cancel{(2,10)}\ (8,6),(10,16),(13,21)\}$

$S^4 = S^3 \cup S_1^3$

$S^4 = \{(0,0),(8,6),(13,16),(13,21)\}$

$(13,21) \in S^4$ & also S^3

~~$S^4=0$, $S^3=1$~~ $x_4 = 0$, $x^3 = 1$

$S_1 \rightarrow x_1$
$S_2 \rightarrow x_2$
$S_3 \rightarrow x_3$
$S_4 \rightarrow x_4$

$(13,21)$
$\downarrow \downarrow$
$(8,6) \rightarrow (5,15) \in S_2 \quad x_2 = 1$
$\downarrow \downarrow$
$(5,15) \rightarrow (0,0) \in S_1 \quad x_1 = 1$

$\in S_0 \quad x_1 = 0$ ✓

$(x_1, x_2, x_3, x_4) \Rightarrow (0,1,1,0)$

∴ The no. of items stored are → 2 i.e item2, item3

Q. Find the max. profit by using dynamic knapsack problem (0/1 knapsack problem) where $m=8$, $n=4$. Each & every item profit & weights are given below

$P = \{1,2,5,6\}$ $\quad P_1 < P_2$

$W = \{2,3,4,5\}$ $\quad w_1 > w_2$

Ans:- $x_1 = 0$
$x_2 = 1$
$x_3 = 0$
$x_4 = 1$

Sol:-

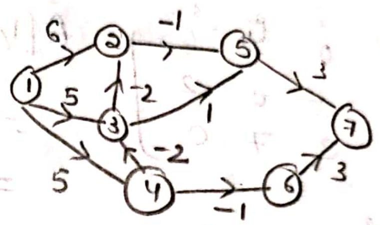
Single Source Shortest path in dynamic programming

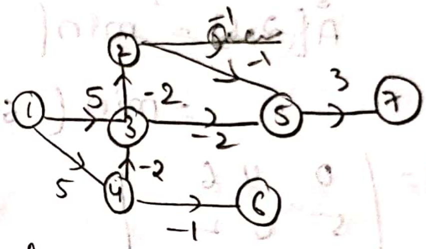
Find out the shortest path from the given graph by using dynamic programming approach. Bellman Ford Alg

sol:-

vertices

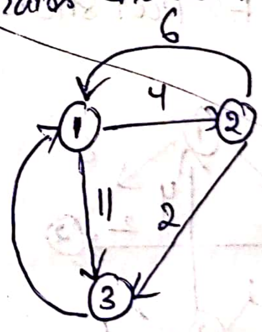
K (edges)	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3
7	0	1	3	5	0	4	3





All pairs shortest path Algorithm

Find out shortest path by using all pairs shortest path Algorithm



$$A^{k}[i,j] = \min\left[A^{k-1}[i,j],\ A^{k-1}[i,k] + A^{k-1}[k,j]\right]$$

here vertices (3) ⇒ we need to calculate A^0, A^1, A^2

$$A^0 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix} \quad A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$A^1[2,3] = \min(A^0[2,3],\ A^0[2,1] + A^0[1,3])$
$= \min(2,\ 6+11)$
$= \{2\}$

$A^1[3,2] = \min(A^0[3,2],\ A^0[3,1] + A^0[1,2])$
$= \min(\infty,\ 3+4) = \{7\}$

$$A^2 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 6 \\ 2 & 6 & 0 & 2 \\ 3 & \cancel{3} & 7 & 0 \end{array}$$

$A^2[1,3] = \min(A^1[1,3], A^1[1,2] + A^1[2,3])$
$= \min(11, 4+2) = 6$

$A^2[3,1] = \min(A^1[3,1], A^1[3,2], A^1[2,1])$
$= \min(3, 7+6) = 3$

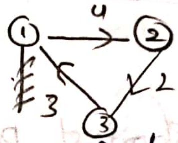
$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^3 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 6 \\ 2 & - & 0 & 2 \\ 3 & 3 & 7 & 0 \end{array}$$

$A^3[1,2] = \min(A^2[1,2], A^2[1,3] + A^2[3,2])$
$= \min(4, 6+7)$
$= 4$

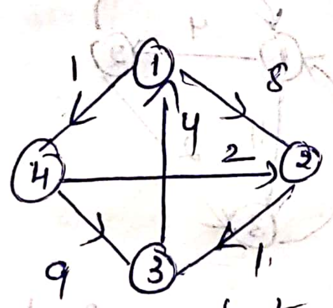
$A^3[2,1] = \min(A^2[2,1], A^2[2,3] + A^2[3,1])$
$= \min(6, 2+3) = 5$

$$A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$



The All pair shortest path matrix is $A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$

(Q.)



Here we have 4 vertices, So, we need to calculate A^1, A^2, A^3, A^4

$$A^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & \infty & 0 & \infty \\ 4 & \infty & 2 & 9 & 0 \end{array}$$

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \infty & 2 & 9 & 0 \end{array}$$

$A^1[2,3] = \min(A^0[2,3], A^0[2,1] + A^0[1,3])$
$= \min(1, \infty + \infty) = 1$

$A^1[2,4] = \min(A^0[2,4],\ A^0[2,1] + A^0[1,4])$
$= \min(\infty,\ \infty + \ ) = \infty$

$A^1[3,2] = \min(A^0[3,2],\ A^0[3,1] + A^0[1,2])$
$= \min(\infty,\ 4+8) = 12$

$A^1[3,4] = \min(A^0[3,4],\ A^0[3,1] + A^0[1,4])$
$= \min(\infty,\ 4+1) = 5$

$A^1[4,2] = \min(A^0[4,2],\ A^0[4,1] + A^0[1,2])$
$= \min(2,\ \infty + 8) = 2$

$A^1[4,4] = \min(A^0[4,4],\ A^0[4,1] + A^0[1,4])$
$= \min(0,\ \infty + 1) = 0$

$A^1[4,3] = \min(A^0[4,3],\ A^0[4,1] + A^0[1,3]) = \min(9,\ \infty + \infty) = 9$

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \infty & 2 & 9 & 0 \end{array} \qquad A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 9 & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \infty & 2 & 3 & 0 \end{array}$$

$A^2[1,3] = \min(A^1[1,3],\ A^1[1,2] + A^1[2,3]) = \min(\infty,\ 8+1) = 9$

$A^2[1,4] = \min(A^1[1,4],\ A^1[1,2] + A^1[2,4]) = \min(1,\ 8+) = 1$

$A^2[3,1] = \min(A^1[3,1],\ A^1[3,2] + A^1[2,1]) = \min(4,\ 12+) = 4$

$A^2[3,4] = \min(A^1[3,4],\ A^1[3,2] + A^1[2,4]) = \min(5,\ 12+) = 5$

$A^2[4,1] = \min(A^1[4,1],\ A^1[4,2] + A^1[2,1]) = \min(\infty,\ 2+\infty) = \infty$

$A^2[4,3] = \min(A^1[4,3],\ A^1[4,2] + A^1[2,3]) = \min(9,\ 2+1) = 3$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 9 & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \infty & 2 & 3 & 0 \end{array} \qquad A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 9 & 1 \\ 2 & 5 & 0 & 1 & 6 \\ 3 & 4 & 12 & 0 & 5 \\ 4 & 7 & 2 & 3 & 0 \end{array}$$

$A^3[1,2] = \min(A^2[1,2],\ A^2[1,3] + A^2[3,2]) = \min(8,\ 9+) = 8$

$A^3[1,4] = \min(A^2[1,4],\ A^2[1,3] + A^2[1,4]) = \min(1,\ 9+) = 1$

$A^3[2,1] = \min(A^2[2,1],\ A^2[2,3] + A^2[3,1]) = \min(\infty,\ 1+4) = 5$

$A^3[2,4] = \min(A^2[2,4],\ A^2[2,3] + A^2[3,4]) = \min(\infty,\ 1+5) = 6$

$A^3[4,1] = \min(A^2[4,1],\ A^2[4,3] + A^2[3,1]) = \min(\infty,\ 3+4) = 7$

Scanned with CamScanner

$A^3[4,2] = \min(A^2[4,2], A^2[4,3] + A^2[3,2])$
$= \min(2, 3+) = 2$

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 9 & 1 \\ 2 & 5 & 0 & 1 & 6 \\ 3 & 4 & 12 & 0 & 5 \\ 4 & 7 & 2 & 3 & 0 \end{array} \qquad A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & - & 1 \\ 2 & - & 0 & - & 6 \\ 3 & - & - & 0 & 5 \\ 4 & 7 & 2 & 3 & 0 \end{array}$$

$A^4[1,2] = \min(A^3[1,2], A^3[1,4] + A^3[4,2]) = \min(8, 1+2) = 3$

$A^4[1,3] = \min(A^3[1,3], A^3[1,4] + A^3[4,3]) = \min(9, 1+3) = 4$

$A^4[2,1] = \min(A^3[2,1], A^3[2,4] + A^3[4,1]) = \min(5, 6+7) = 5$

$A^4[2,3] = \min(A^3[2,3], A^3[2,4] + A^3[4,3]) = \min(1, 6+) = 1$

$A^4[3,1] = \min(A^3[3,1], A^3[3,4] + A^3[4,1]) = \min(4, 5+) = 4$

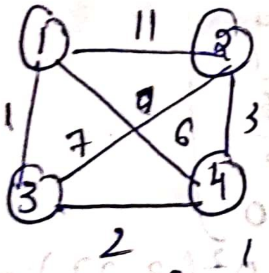
$A^4[3,2] = \min(A^3[3,2], A^3[3,4] + A^3[4,2]) = \min(12, 5+2) = 7$

$$A^4 = \begin{bmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix}$$

∴ The All pairs shortest path matrix is

$$A^4 = \begin{bmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix}$$

(3)



Here vertices are 4
so, to find A^1, A^2, A^3, A^4

$$A^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array} \qquad A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & - & - \\ 3 & 1 & - & 0 & - \\ 4 & 6 & - & - & 0 \end{array}$$

$=\min(8, 1+2)=3$

$3)=4$

$+7)=5$

$+)=1$

$, 5+)=4$

$12, 5+2)=7$

4

$A^1[2,3] = \min(A^0[2,3], A^0[2,1]+A^0[1,3]) = \min(7, 11+) = 7$

$A^1[2,4] = \min(A^0[2,4], A^0[2,1]+A^0[1,4]) = \min(3, 11+) = 3$

$A^1[3,2] = \min(A^0[3,2], A^0[3,1]+A^0[1,2]) = \min(7, 1+11) = 7$

$A^1[3,4] = \min(A^0[3,4], A^0[3,1]+A^0[1,4]) = \min(2, 1+6) = 2$

$A^1[4,2] = \min(A^0[4,2], A^0[4,1]+A^0[1,2]) = \min(3, 6+11) = 3$

$A^1[4,3] = \min(A^0[4,3], A^0[4,1]+A^0[1,3]) = \min(2, 6+1) = 2$

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array} \qquad A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & - & - \\ 2 & 11 & 0 & 7 & 3 \\ 3 & - & 7 & 0 & - \\ 4 & - & 3 & - & 0 \end{array}$$

$A^2[1,3] = \min(A^1[1,3], A^1[1,2]+A^1[2,3]) = \min(1, 11+) = 1$

$A^2[1,4] = \min(A^1[1,4], A^1[1,2]+A^1[2,4]) = \min(6, 11+) = 6$

$A^2[3,1] = \min(A^1[3,1], A^1[3,2]+A^1[2,1]) = \min(1, 7+) = 1$

$A^2[3,4] = \min(A^1[3,4], A^1[3,2]+A^1[2,4]) = \min(2, 7+) = 2$

$A^2[4,1] = \min(A^1[4,1], A^1[4,2]+A^1[2,1]) = \min(6, 3+11) = 6$

$A^2[4,3] = \min(A^1[4,3], A^1[4,2]+A^1[2,3]) = \min(2, 3+7) = 2$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 11 & 1 & 6 \\ 2 & 11 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array} \qquad A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & - & 1 & - \\ 2 & - & 0 & 7 & - \\ 3 & 1 & 7 & 0 & 2 \\ 4 & - & - & 2 & 0 \end{array}$$

$A^3[1,2] = \min(A^2[1,2], A^2[1,3]+A^2[3,2]) = \min(11, 1+7) = 8$

$A^3[1,4] = \min(A^2[1,4], A^2[1,3]+A^2[3,4]) = \min(6, 1+2) = 3$

$A^3[2,1] = \min(A^2[2,1], A^2[2,3]+A^2[3,1]) = \min(11, 7+1) = 8$

$A^3[2,4] = \min(A^2[2,4], A^2[2,3]+A^2[3,4]) = \min(3, 7+2) = 3$

$A^3[4,1] = \min(A^2[4,1], A^2[4,3]+A^2[3,1]) = \min(6, 2+11) = 6$

$A^3[4,2] = \min(A^2[4,2], A^2[4,3]+A^2[3,2]) = \min(3, 2+7) = 3$

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 1 & 3 \\ 2 & 8 & 0 & 7 & 3 \\ 3 & 1 & 7 & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array} \qquad A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & - & - & 3 \\ 2 & - & 0 & - & 3 \\ 3 & - & - & 0 & 2 \\ 4 & 6 & 3 & 2 & 0 \end{array}$$

$A^4[1,2] = \min(A^3[1,2], A^3[1,4]+A^3[4,2]) = \min(8, 3+3) = 6$

$A^4[1,3] = \min(A^3[1,3], A^3[1,4]+A^3[4,3]) = \min(1, 3+) = 1$

$A^4[2,1] = \min(A^3[2,1], A^3[2,4]+A^3[4,1]) = \min(8, 3+6) = 8$

$A^4[2,3] = \min(A^3[2,3], A^3[2,4]+A^3[4,3]) = \min(7, 3+2) = 5$

$A^4[3,1] = \min(A^3[3,1], A^3[3,4]+A^3[4,1]) = \min(1, 2+6) = 1$

$A^4[3,2] = \min(A^3[3,2], A^3[3,4]+A^3[4,2]) = \min(7, 2+3) = 5$

$$A^4 = \begin{bmatrix} 0 & 6 & 1 & 3 \\ 8 & 0 & 5 & 3 \\ 1 & 5 & 0 & 2 \\ 6 & 3 & 2 & 0 \end{bmatrix}$$ is req matrix of all pairs shortest path.

* **Travelling Salesman problem**

Find out a shortest path by using travelling salesman problem in dynamic approach

e.g:- Formula:-

$$g(i,S) = \min_{j \in S} \{ C_{ij} + g(j, S - \{j\}) \}$$

e.g:-

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 15 & 20 \\ 2 & 5 & 0 & 9 & 10 \\ 3 & 6 & 13 & 0 & 12 \\ 4 & 8 & 8 & 9 & 0 \end{array}$$

Initially $g(i,\phi) = C_{i1}$

$|S| = 0$

$g(1,\phi) = C_{11} = C_{11} = 0$

$g(2,\phi) = C_{21} = 5$

$g(3,\phi) = C_{31} = 6$

$g(4,\phi) = C_{41} = 8$

The set contain 1 element i.e $|S| = 1$

$g(2, \{3\}) = \min\{ C_{ij} + g(j, S - \{j\}) \}$

$g(2, \{4\}) = \min\{ C_{23} + g(3, \{3\} - \{3\}) \}$

$g(3, \{2\}) = \min\{ C_{23} + g\{3, \phi\} \}$

$g(3, \{4\}) = \min\{ 9 + 6 \}$

$g(4, \{2\}) = 15$

$g(4, \{3\})$

$g(2, \{4\}) = \min$

$g(3, \{2\}) = \min$

$g(3, \{4\}) = \min$

$g(4, \{2\}) =$

$g(4, \{3\}) =$

set contains

$g(2, \{3,4\})$

$g(3, \{2,4\})$

$g(4, \{2,3\})$

set contain

$g(1, \{2,3,4$

$1 \xrightarrow{10} 2$

(2) Find o

for the

	A	B
A	0	12
B	12	0
C	10	3
D	19	
E	8	

Initiall

$|S| = 0$

$g(A, \phi$

$g(B,$

$g(2, \{4\}) = \min\{c_{24} + g(4, \{4\} - \{4\})\}$
$= \min\{10 + g(4, \phi)\} = 10+8 = 18$

$g(3, \{2\}) = \min\{c_{32} + g(2, \{2\} - \{2\}\} = 13+5 = 18$

$g(3, \{4\}) = \min\{c_{34} + g(4, \{4\} - \{4\}\} = 12+8 = 20$

$g(4, \{2\}) = \min\{c_{42} + g(2, \{2\} - \{2\}\} = 8+5 = 13$

$g(4, \{3\}) = \min\{c_{43} + g(3, \{3\} - \{3\})\} = 9+6 = 15$

set contains two vertices $|S| = 2$

$g(2, \{3,4\}) = \min\begin{cases} c_{23} + g(3, \{3,4\} - \{3\}) \\ c_{24} + g(4, \{3,4\} - \{4\}) \end{cases} = \begin{cases} 9+20 = 29 \\ 10+15 = 25 \end{cases} = 25$

$g(3, \{2,4\}) = \min\begin{cases} c_{32} + g(2, \{2,4\} - \{2\}) = 13+18 = 31 \\ c_{34} + g(4, \{2,4\} - \{4\}) = 12+13 = 25 \end{cases} = 25$

$g(4, \{2,3\}) = \min\begin{cases} c_{42} + g(2, \{2,3\} - \{2\}) = 8+15 = 23 \\ c_{43} + g(3, \{2,3\} - \{3\}) = 9+18 = 27 \end{cases} = 23$

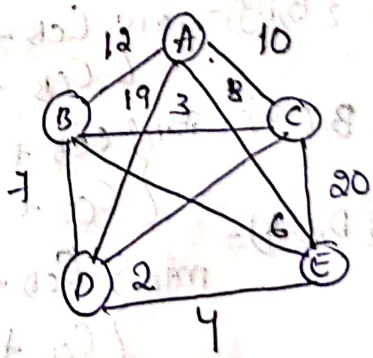
set contains three vertices $|S| = 3$

$g(1, \{2,3,4\}) = \min\begin{cases} c_{12} + g(2, \{2,3,4\} - \{2\}) = 10+25 = 35 \\ c_{13} + g(3, \{2,3,4\} - \{3\}) = 15+25 = 40 \\ c_{14} + g(4, \{2,3,4\} - \{4\}) = 20+23 = 43 \end{cases}$

$1 \xrightarrow{10} 2 \xrightarrow{10} 4 \xrightarrow{9} 3 \xrightarrow{6} 1 \Rightarrow 10+10+9+6 = 35$

② Find out a shortest path for travelling salesman problem for the given graph.

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 12 & 10 & 19 & 8 \\ B & 12 & 0 & 3 & 7 & 6 \\ C & 10 & 3 & 0 & 2 & 20 \\ D & 19 & 7 & 2 & 0 & 4 \\ E & 8 & 6 & 20 & 4 & 0 \end{array}$$



Initially $g(1, \phi) = C_{1A}$

$|S| = 0$

$g(A, \phi) = 0$

$g(B, \phi) = 12$

$g(C, \phi) = 10$

$g(D, \phi) = 19$

$g(E, \phi) = 8$

The set contain one element $|S| = 1$

$g(B, \{C\}) = \min(C_{BC}, g(C, \{C\} - \{C\}) = 3+10 = 13$

$g(B, \{D\}) = \min(C_{BD}, g(D, \{D\} - \{D\}) = 3+19 = 22$

$g(B, \{E\}) = \min(C_{BE}, g(E, \{E\} - \{E\}) = 6+8 = 14$

$g(C, \{B\}) = \min(C_{CB}, g(B, \{B\} - \{B\}) = 3+12 = 15$

$g(C, \{D\}) = \min(C_{CD}, g(D, \{D\} - \{D\}) = 2+19 = 21$

$g(C, \{E\}) = \min(C_{CE}, g(E, \{E\} - \{E\}) = 20+8 = 28$

$g(D, \{B\}) = \min(C_{DB}, g(B, \{B\} - \{B\})) = 7+12 = 19$

$g(D, \{C\}) = \min(C_{DC}, g(C, \{C\} - \{C\}) = 2+10 = 12$

$g(D, \{E\}) = \min(C_{DE}, g(E, \{E\} - \{E\}) = 4+8 = 12$

$g(E, \{B\}) = \min(C_{EB}, g(B, \{B\} - \{B\}) = 6+12 = 18$

$g(E, \{C\}) = \min(C_{EC}, g(C, \{C\} - \{C\}) = 20+10 = 30$

$g(E, \{D\}) = \min(C_{ED}, g(D, \{D\} - \{D\}) = 4+19 = 23$

The set contain 2 elements $|S| = 2$

$g(B, \{C, D\}) = \min\begin{cases} C_{BC} + g(C, \{C,D\} - \{C\}) \\ C_{BD} + g(D, \{C,D\} - \{D\}) \end{cases}$

$g(B, \{D, E\}) = \min\begin{cases} C_{BD} + g(D, \{D,E\} - \{D\}) \\ C_{BE} + g(E, \{D,E\} - \{E\}) \end{cases}$

$g(B, \{C, E\}) = \min\begin{cases} C_{BC} + g(C, \{C,E\} - \{C\}) \\ C_{BE} + g(E, \{C,E\} - \{E\}) \end{cases}$

$g(C, \{B, D\}) = \min\begin{cases} C_{CB} + g(B, \{B,D\} - \{B\} \\ C_{CB} + g(B, \{B,D\} - \{D\} \end{cases}$

$g(C, \{B, E\}) = \min\begin{cases} C_{CB} + g(\\ C_{CE} + \end{cases}$

$g(C, \{D, E\}) = \min\begin{cases} C_{CD} + \\ C_{CE} + \end{cases}$

$g(D, \{B, C\}) = \min\begin{cases} C_{DB} + \\ C_{DC} + \end{cases}$

$g(D, \{B, E\}) = \min\begin{cases} C_{DB} + \\ C_{DE} + \end{cases}$

$g(D, \{C, E\}) = m$

$g(E, \{B, C\}) =$

$g(E, \{B, D\}) =$

$g(E, \{C, D\}$

Back Tra

Sum of

$\Rightarrow m = 5$

are . C w

$x_4 = 1$

6, 5, 0

The

0 1

The set contain one element $|S| = 1$

$g(B, \{C\}) = \min(C_{BC}, g(C, \{C\} - \{C\}) = 3 + 10$

$g(B, \{D\}) = \min(C_{BD}, g(D, \{D\} - \{D\}) = 3 + 19 = 22$

$g(B, \{E\}) = \min(C_{BE}, g(E, \{E\} - \{E\}) = 6 + 8 = 14$

$g(B, \{E\}) = \min(C_{BE}, g(E, \{E\} - \{E\}) = 3 + 12 = 15$

$g(C, \{B\}) = \min(C_{CB}, g(B, \{B\} - \{B\}) = 3 + 12 = 15$

$g(C, \{B\}) = \min(C_{CB}, g(B, \{B\} - \{B\}) = 2 + 19 = 21$

$g(C, \{D\}) = \min(C_{CD}, g(D, \{D\} - \{D\}) = 20 + 8 = 28$

$g(C, \{E\}) = \min(C_{CE}, g(E, \{E\} - \{E\}) = 7 + 12 = 19$

$g(D, \{B\}) = \min(C_{DB}, g(B, \{B\} - \{B\})) = 7 + 12 = 19$

$g(D, \{C\}) = \min(C_{DC}, g(C, \{C\} - \{C\}) = 2 + 10 = 12$

$g(D, \{E\}) \min(C_{DE}, g(E, \{E\} - \{E\}) = 4 + 8 = 12$

$g(E, \{B\}) = \min(C_{EB}, g(B, \{B\} - \{B\}) = 6 + 12 = 18$

$g(E, \{C\}) = \min(C_{EC}, g(C, \{C\} - \{C\}) = 20 + 10 = 30$

$g(E, \{D\}) = \min(C_{ED}, g(D, \{D\} - \{D\}) = 4 + 17 = 23$

The set contain 2 elements $|S| = 2$

$$g(B, \{C, D\}) = \min\begin{cases} C_{BC} + g(C, \{C, D\} - \{C\}) \\ C_{BD} + g(D, \{C, D\} - \{D\}) \end{cases}$$

$$g(B, \{D, E\}) = \min\begin{cases} C_{BD} + g(D, \{D, E\} - \{D\}) \\ C_{BE} + g(E, \{D, E\} - \{E\}) \end{cases}$$

$$g(B, \{C, E\}) = \min\begin{cases} C_{BC} + g(C, \{C, E\} - \{C\}) \\ C_{BE} + g(E, \{C, E\} - \{E\}) \end{cases}$$

$$g(C, \{B, D\}) = \min\begin{cases} C_{CB} + g(B, \{B, D\} - \{B\} \\ C_{CB} + g(B, \{B, D\} - \{D\} \end{cases}$$

$$g(C, \{B, E\}) = \min\begin{cases} C_{CB} + g(\\ C_{CE} + \end{cases}$$

$$g(C, \{D, E\}) = \min\begin{cases} C_{CD} + \\ C_{CE} + \end{cases}$$

$$g(D, \{B, C\}) = \min\begin{cases} C_{DB} + \\ C_{DC} + \end{cases}$$

$$g(D, \{B, E\}) = \min\begin{cases} C_{DB} + \\ C_{DE} + \end{cases}$$

) = 3+10 = 13
9 = 22
= 19
= 15
21
28
19
= 12
12
18
30
23

$g(D, \{C,E\}) = \min\begin{cases} C_{DC} \\ C_{DE} \end{cases}$

$g(E, \{B,C\}) = \min\begin{cases} C_{EB} \\ C_{EC} \end{cases}$

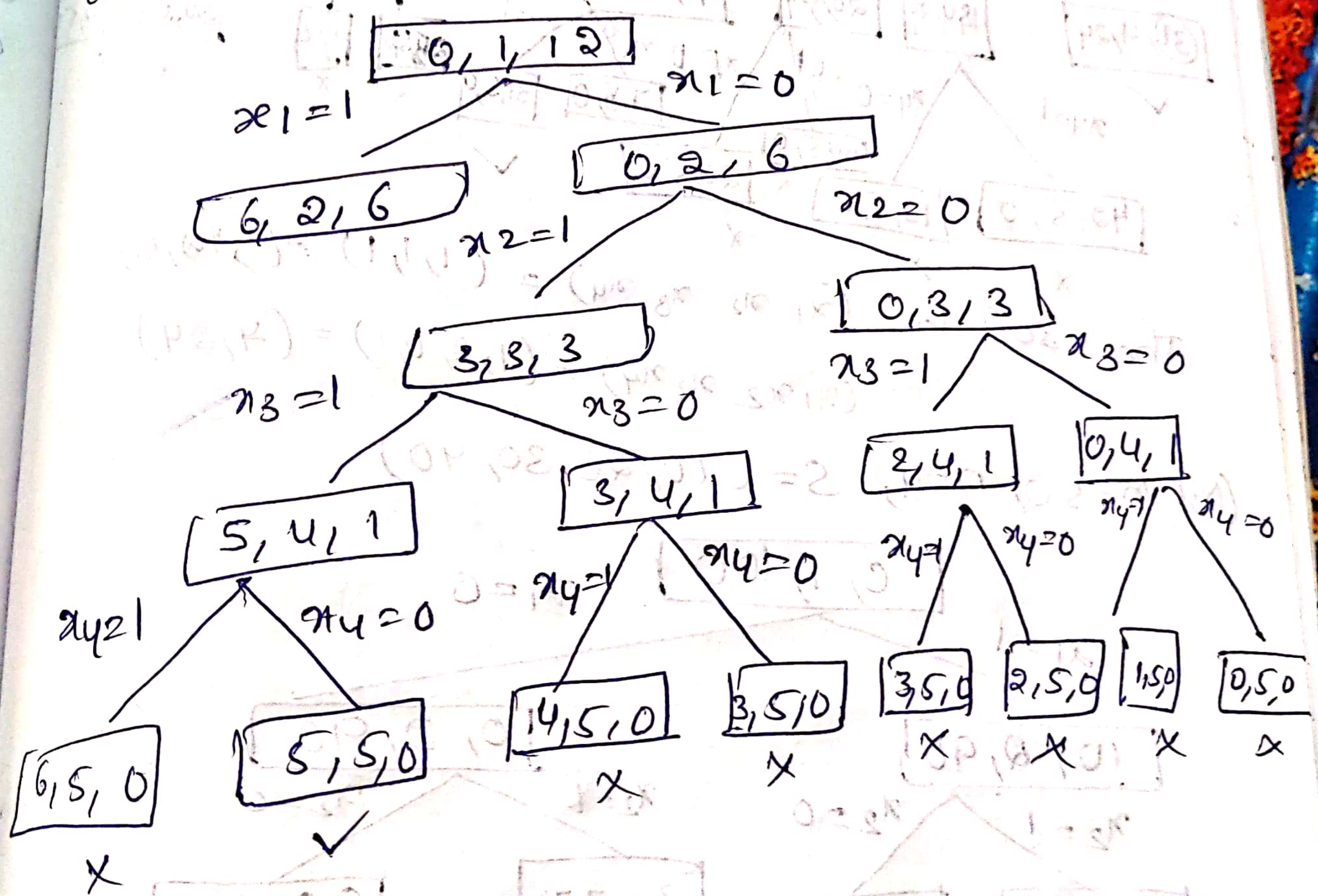
$g(E, \{B,D\}) = \min\begin{cases} C_{EB} \\ C_{ED} \end{cases}$

$g(E, \{C,D\}) = \min\begin{cases} C_{EC} \\ C_{ED} \end{cases}$

Back Tracking

Sum of Subsets :-

$\Rightarrow$ $m=5$, $n=4$ the weights of the objects are $(w_1, w_2, w_3, w_4) = (6,3,2,1)$

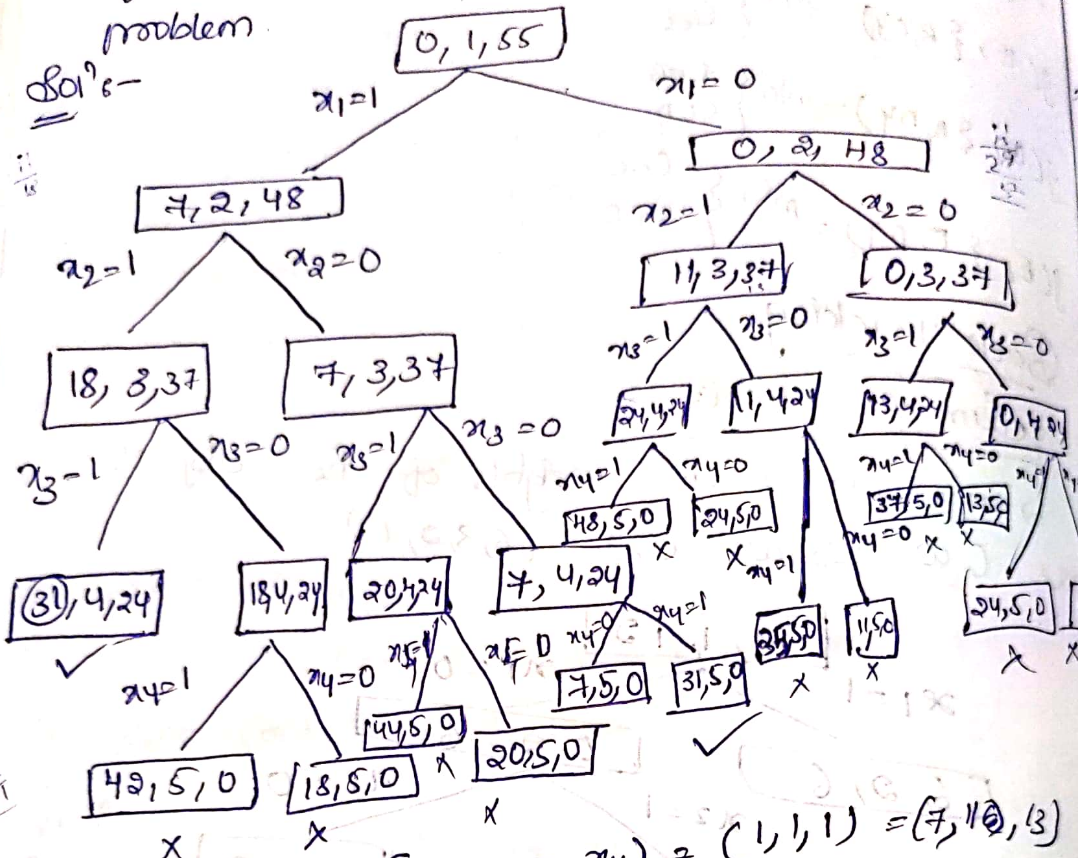


x_1	x_2	x_3	x_4
0	1	1	0

The feasible sol^n is $x_1\ x_2\ x_3\ x_4$ is 0 1 1 0. The sol^n is 3 2

(2) $n=4$, $(w_1, w_2, w_3, w_4) = (7, 11, 13, 24)$ where $m=31$ find out all possible sol's by using sum of subset problem

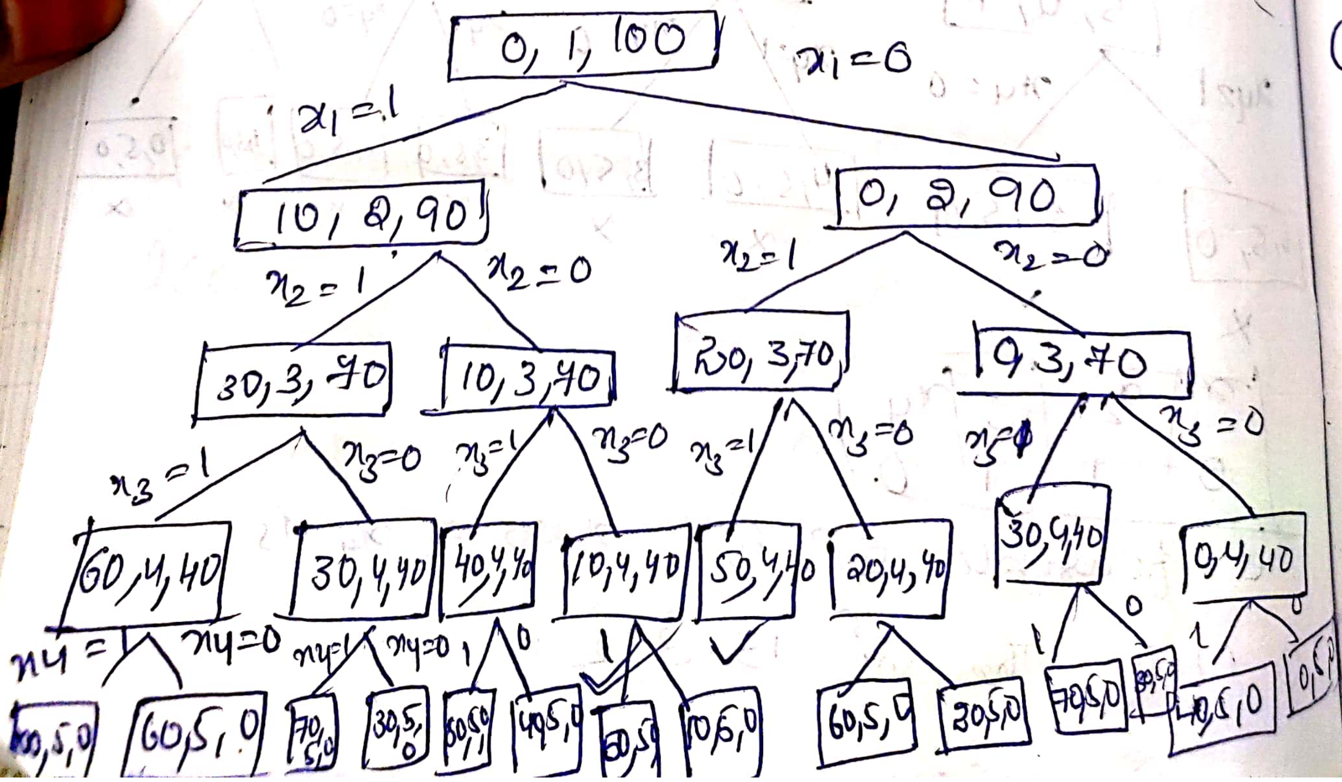
Sol:-



The sol is $(x_1\ x_2\ x_3\ x_4) = (1, 1, 1) = (7, 11, 13)$

$(x_1 x_2 x_3 x_4) = (1, 0, 0, 1) = (7, 24)$.

(Q.) $m=50$, $n=4$, $S=(10, 20, 30, 40)$



The sol is ...

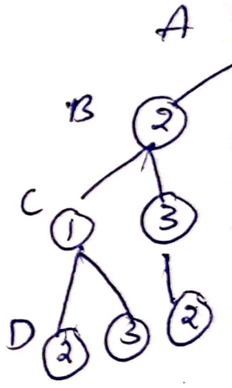
Greedy Alg

Real-life Applica...

Graph col...

(Q.) No-of noc...
No-of Col...
by using ...
graph. Th...

Sol:- $m=3$



(Q.)

the soln is $(x_1, x_2, x_3, x_4) = (0,1,1,\ldots) = (90, 80)$

$(x_1\ x_2\ x_3\ x_4) = (1,0,0,1) = (10, 40)$

Greedy Alg

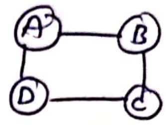
real-life Applications

Graph colouring problem

(Q) No. of nodes = 4

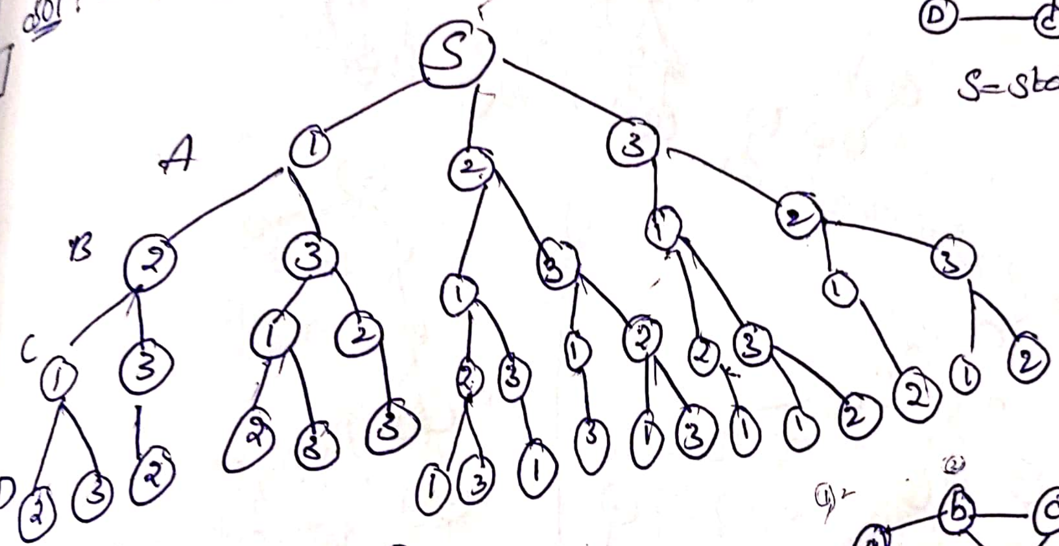
No. of Colors = 3 those are m = (1, 2, 3)

by using of graph colouring problem color the graph. The graph is given below.

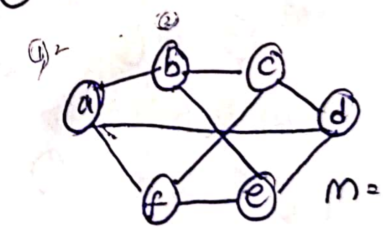


soln:- m = 3 (1, 2, 3)

S = start

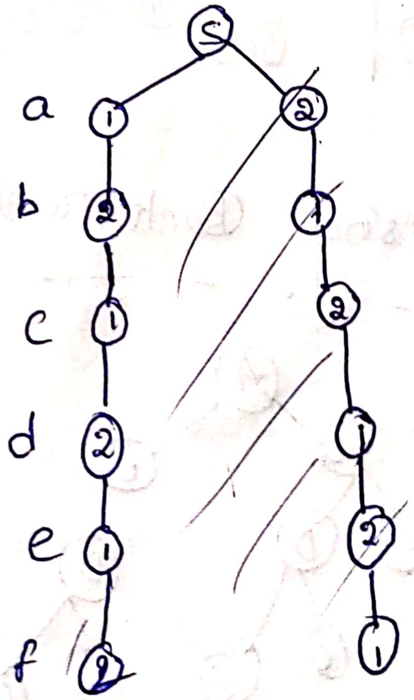


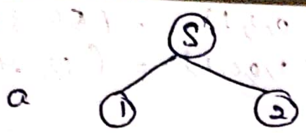
Q)

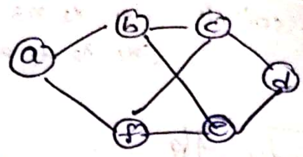


m = 2

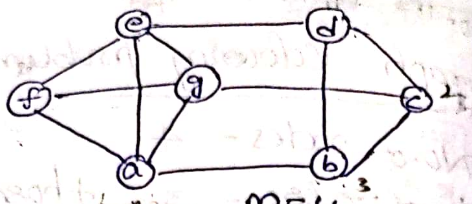
(Q.)

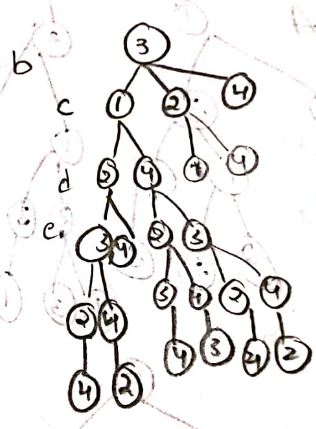






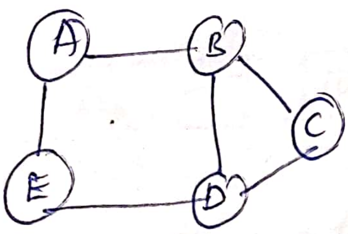
(Q.)

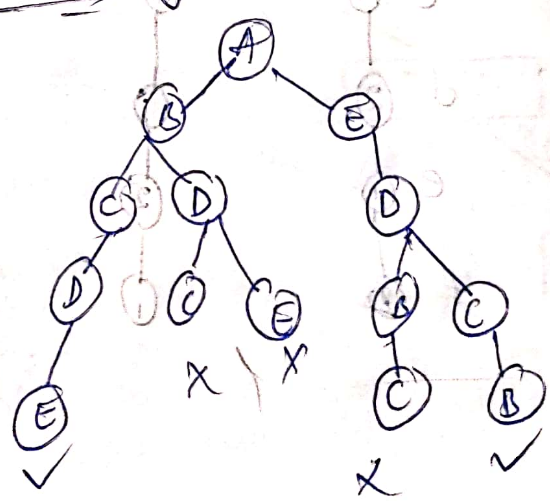




13/03/23 Hamilton cycle using Back-tracking

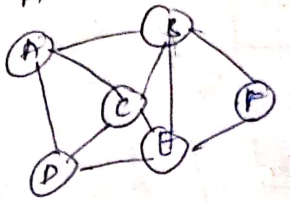
→

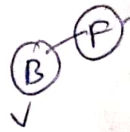




∴ Possible hamilto
A → B → C →
A → E → D →

→



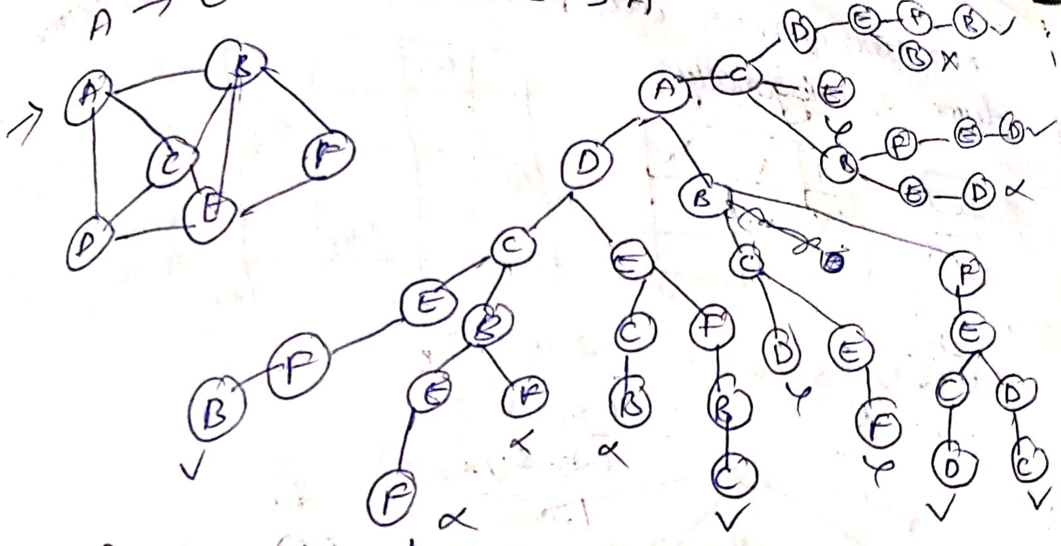


∴ The possi
A B F E
A B F E
A
A

→ Find out

∴ A → B → C → D → E → A

A → E → D → C → B → A

→ 

∴ The possible hamilton paths are

A B F E C D

A B F E D C

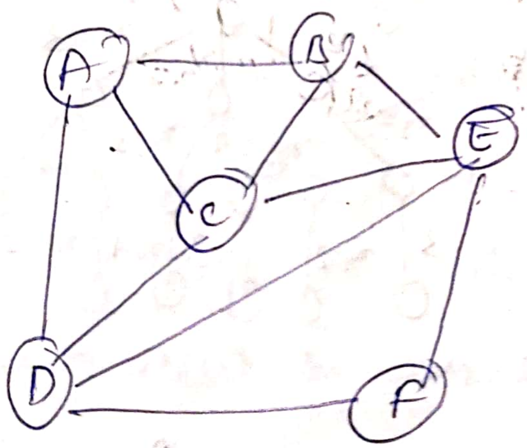
A C B F E D

A C D E F B

A D C E F B

A D E F B C

→ Find out hamiltanion paths for following graph.



The possible paths are

A B C E F D

A B E F D C A

A C B E F D

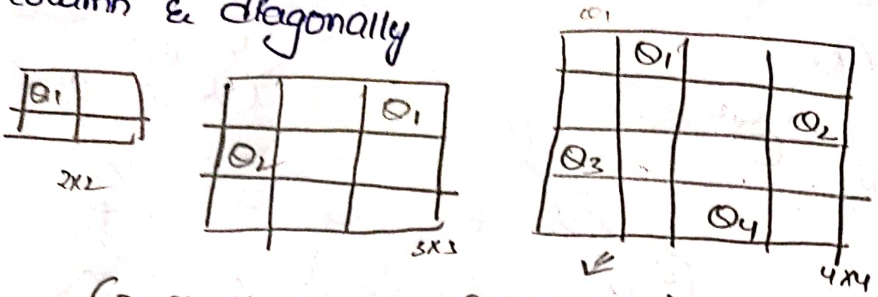
A C D F E B

A D F E C B

A D F E B C

N-Queens Problem

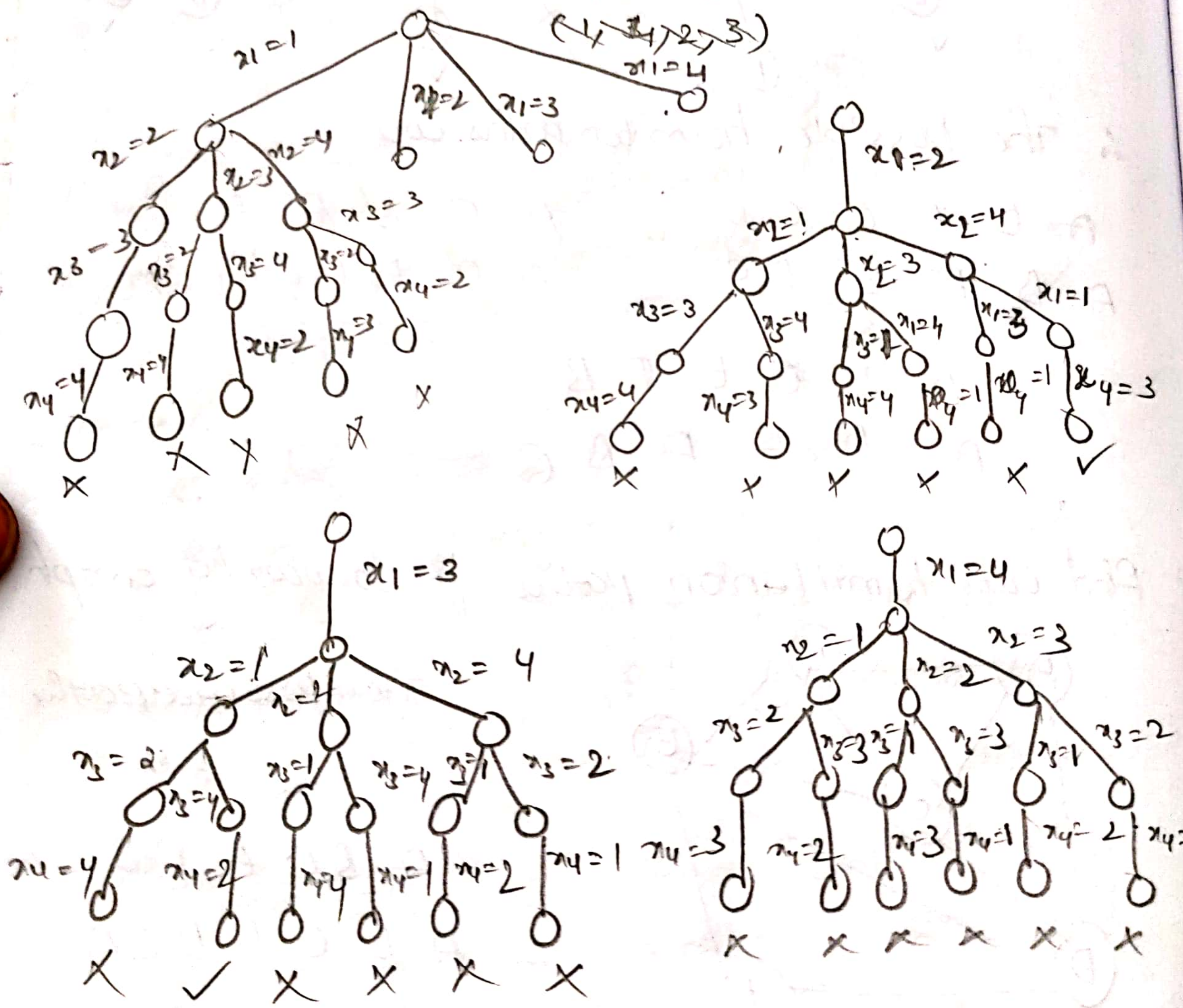
No two queens are placed in same row, same column & diagonally



This is only a sol^n. There are many sol's. Backtracking is using here.

$(x_1, x_2, x_3, x_4) = (2, 4, 1, 3)$

$(3, 1, 4, 2)$



∴ The sol^n are or the feasible sol's are

$(Q_1, Q_2, Q_3, Q_4) = (2, 4, 1, 3)$

$(3, 1, 4, 2)$ //.

V. Branch & Bound

0/1 knapsack problem

1. Least-cost
2. FIFO

0/1 knapsack by using Least-cost

Q. $n=4$, $(P_1, P_2, P_3, P_4) = (10, 10, 12, 18)$, $(w_1, w_2, w_3, w_4) = (2, 4, 6, 9)$, $m = 15$

Sol:- Inorder to calculate branch & bound minimization problem we are going to construct state space tree for 0/1 knapsack problem.

Branch & Bound is optimization problem so we are taking a -ve profits for the given knapsack problem.

calculate upper bound & lower bound for each & every state for each & every state

→ The lowerbound is represented by using $\hat{c}$

The upperbound is represented by using $\hat{U}$

The lowerbound means those can be accepted the fractional values.

upperbound doesn't ~~repre~~ accepted the fractional values.

Sol:- Given (P_1, P_2, P_3, P_4) (w_1, w_2, w_3, w_4)

$(10, 10, 12, 18)$ $(2, 4, 6, 9)$

$x_1 \quad x_2 \quad x_3 \quad x_4$

$n=4$, $m=15$

$(w_1, w_2, w_3, w_4) =$

minimization

take space

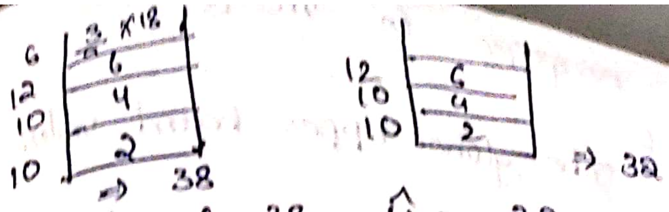
n 80

given

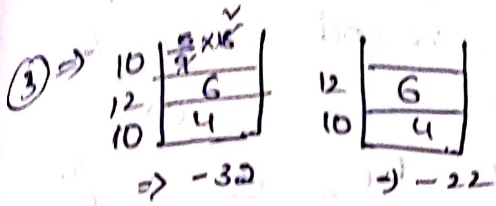
h & every

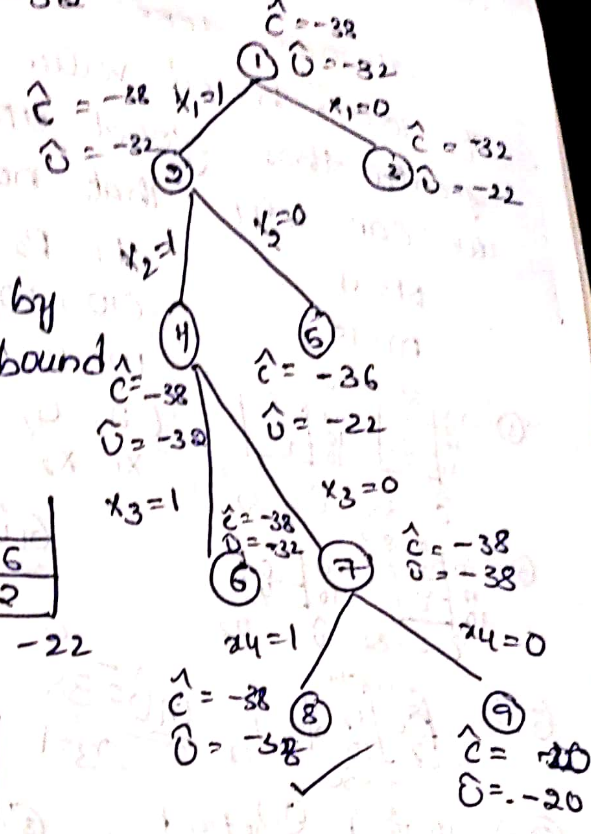
ted

e

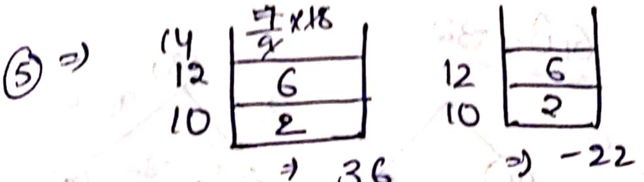


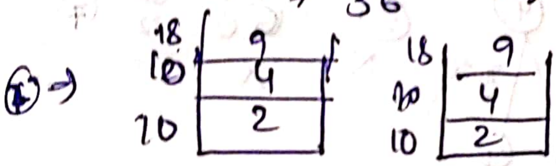
$\hat{c} = -38, \quad \hat{u} = -32$

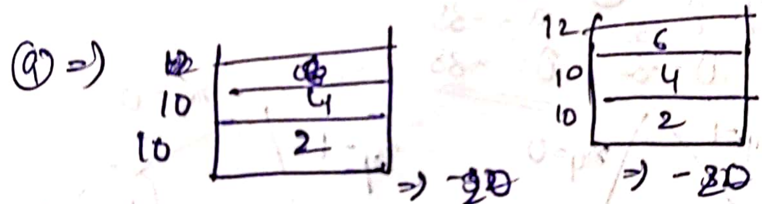




we can expand the tree by considering less lower bound value of the node.







$(x_1, x_2, x_3, x_4) = (1, 1, 0, 1)$

$\Rightarrow -10 - 10 + 0 - 18 \Rightarrow -38$

The minimization problem using Least cost Branch & bound is -38

$\therefore$ max profit using 0/1 knapsack = 38.

0/1 knapsack by using FIFO

Q. Find the max profit of 0/1 knapsack problem by using FIFO branch & bound when no. of objects n = 4, profit & wgts of corresponding objects are P(10, 10, 12, 18)

W(2, 4, 6, 9)

The knapsack bagsize is $m = 15$.

Sol^n :-

Note :- The global upper bound value is always less.

If lower bound value of particular node is greater than global upper bound value then we can kill that node.

$N = 4$

$m = 15$

	P_1	P_2	P_3	P_4
	(10,	10,	12,	18)
	w_1	w_2	w_3	w_4
	(2,	4,	6,	9)
	x_1	x_2	x_3	x_4

① ⇒ 18, 12, 10, 14; $3/9 \times 18 = 6$; 12, 10, 10

$\hat{c} = -38$ $\quad \hat{u} = -32$

③ ⇒ 10, 12, 10; $5 \times 18 / 9$; 12, 10

$\hat{c} = -32$ $\quad \hat{u} = -22$

⑤ ⇒ $7 \times 18 / 9 = 14$; 12, 10 $= -36$; 12, 10 $= -22$

⑥ ⇒ 10, 12, 10 ; $8 \times 18 / 9$

⑦ ⇒ 18, 12

⑨ ⇒ 12, 10, 10

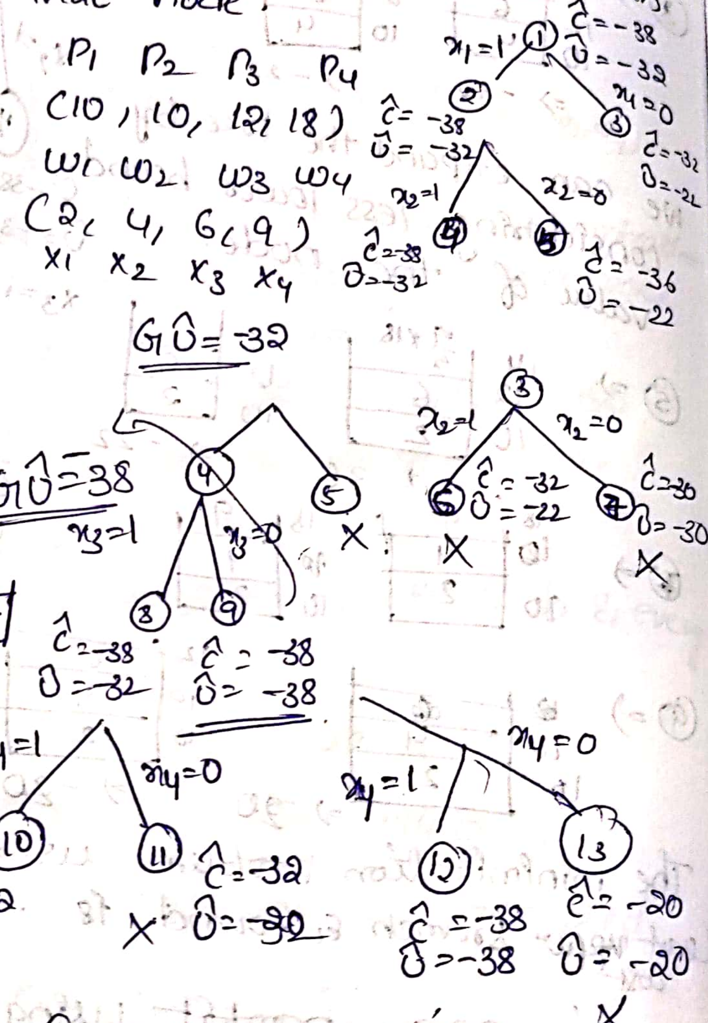
⑪ ⇒ 12, 10, 10

⑫ ⇒ 18, 10, 10

⑬ ⇒ 10, 10

$G\hat{U} = -32$

$G\hat{U} = -38$



$(x_1, x_2, x_3, x_4) = (1, 1, 0, 1) \Rightarrow -10 - 10 + 0 - 18 = -38$

Max profit using FIFO $= 10 + 10 + 18 = 38$.

Scanned with CamScanner

Travelling SellsPerson Problem :- (using branch & bound)

Find the shortest path for the given graph of travelling salesperson problem by using branch & bound technique.

$$\begin{array}{c} -10 \\ -2 \\ -2 \\ -3 \\ -4 \end{array}\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

check the given graph is in reduced matrix or not. If the matrix is not reduced. convert into reduced matrix.

A reduced matrix means in each row & column the matrix contains atleast one zero if the matrix is not in a reduced matrix first we do the row reduction & column reduction

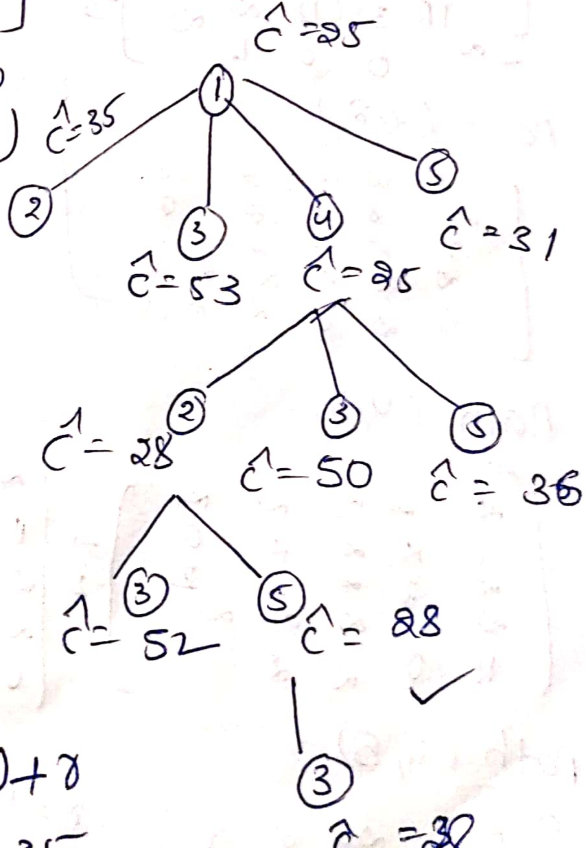
$$\Rightarrow \begin{bmatrix} \infty & 10 & \cancel{20} & \cancel{30} & \cancel{10} & \cancel{11} \\ 15 & \cancel{\infty} & \cancel{16} & 4 & 2 \end{bmatrix}$$

$$\begin{array}{ccccc} -1 & & -3 & & \end{array}$$
$$\begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

$$\Rightarrow A = \begin{bmatrix} \infty & 10 & \cancel{17} & 1 \\ 12 & \infty & 11 & 0 \\ 0 & 3 & \infty & 2 \\ 15 & 3 & 12 & 0 \\ 11 & 0 & 0 & \infty \end{bmatrix}$$

$r = 10+2+2+3+4+1+3$
$= 25$

Rule:
1) The path is (i,j) than place ∞ in row i $\infty = \begin{pmatrix} \text{row } i \\ \text{column } j \end{pmatrix}$
2) and also place ∞ in $(j,1)$



path (1,2) :-

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & 2 & 0 \\ 0 & \infty & \infty & 0 & 2 \\ 15 & \infty & 12 & \infty & 0 \\ 11 & \infty & 0 & 12 & \infty \end{bmatrix}$$

$r = 0$

parent node

$\hat{c}(2) = \hat{c}(1) + \hat{c}(1,2) + r$
$= 25 + 10 + 0 = 35$

Path (1,3)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & \infty & 2 & 0 \\ \infty & 3 & \infty & 0 & 2 \\ 15 & 3 & \infty & \infty & 0 \\ 11 & 0 & \infty & 12 & \infty \end{bmatrix} \Rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & \infty & 2 & 0 \\ \infty & 3 & \infty & 0 & 2 \\ 4 & 3 & \infty & \infty & 0 \\ 0 & 0 & \infty & 12 & \infty \end{bmatrix}$$

$\gamma = 11$

$\hat{c}(3) = \hat{c}(1) + c(1,3) + \gamma = 25 + 17 + 11 = 53$

⇒ path (1,4)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ \infty & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & \infty & \infty \end{bmatrix}$$

$\gamma = 0$

$\Rightarrow \hat{c}(4) = \hat{c}(1) + c(1,4) + \gamma$
$= 25$

path (1,5)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & 2 & \infty \\ 0 & 3 & \infty & 0 & \infty \\ 15 & 3 & 12 & \infty & \infty \\ \infty & 0 & 0 & 12 & \infty \end{bmatrix} \begin{matrix} \\ -2 \\ \\ -3 \\ \\ \end{matrix}$$

$\gamma = 5$

$\Rightarrow \hat{c}(5) = \hat{c}(1) + c(1,5) + \gamma$
$= 25 + 1 + 5$
$= \cancel{26}\ 30$

⇒ path (4,2)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & 0 & \infty & \infty \end{bmatrix}$$

$\gamma = 0$

$\hat{c}(2) = \hat{c}(4) + c(4,2) + \gamma$
$= 25 + 3$
$= 28$

path (4,3)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & \infty & \infty & 0 \\ \infty & 3 & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & 0 & \infty & \infty & \infty \end{bmatrix} \begin{matrix} \\ \\ -2 \\ \\ \\ \end{matrix} \Rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & \infty & \infty & 0 \\ \infty & 1 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \end{bmatrix}$$

$\gamma = 13$

$\Rightarrow \hat{c}(3) = \hat{c}(4) + c(4,3) + \gamma$
$= 25 + 12 + 13$
$= 50$

path (4,5)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & \infty \\ 0 & 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty & \infty \end{bmatrix} -11$$

$\gamma = 11$

$\hat{c}(5) = \hat{c}(4) + \hat{c}(4,5) + \gamma$
$= 25 + 0 + 11$
$= 36$

path (2,3)

path (2,5)

The sho

~~Total cos~~

cost =

⇒ String

Find

string

s101

s102

Ansr

path (2,3)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & \infty & \infty & \infty \end{bmatrix} \begin{matrix} -2 \\ \\ \\ \\ -11 \end{matrix}$$

$r = 13$

$\hat{c}(3) = \hat{c}(2) + \hat{c}(2,3) + r$

$= 28 + 13 + 11$

$= 52$

path (2,5)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$\hat{c}(5) = \hat{c}(2) + \hat{c}(2,5) + r$

$= 28 + 0 + 0$

$= 28$

path (5,3)

$\hat{c} = 28$

The shortest path is 1 → 4 → 2 → 5 → 3

~~Total cost = 25+28+28+28 =~~

cost = 28.

⇒ String - Editing Problem

Find the no. of operations are req by using string editing problem. The strings are given below.

str1 = abcfg

str2 = adceg

1) Insert
2) Remove
3) Replace.

Ans:-

	Null	a	b	c	f	g
Null	0	1	2	3	4	5
a	1	0	1	2	3	4
d	2	1	1	2	3	4
c	3	2	2	1	2	3
e	4	3	3	2	2	3
g	5	4	4	3	3	2

Rule 1 :- If row value ≠ column value (r ≠ c) then

min +1

Rule 2 :- If r = c then

copy

a b c f g

a d c e g

Str3 = ab[illegible]ceg

←

→ ⟺ Insert operation

$\boxed{\downarrow}$ ⟺ Replace Operation

Total no. of operations = 2