[redacted]

[redacted]

4. Dynamic programming

* Dynamic programming is an algorithm x that paradist solves a given complex problem by breaking it into sub problems and storing the results of the sub problems to avoid recom-puting of the subproblems.

* storing the values into the table and using them later is also called as programming.

* Before computing a function we check the table if the table is empty, then only dynamically calling the function.

when to use dynamic programming:

If the problem satisfies the following properties:

* overlapping the sub problems
* optimal substructure (or) prinipal of optimality (pop)

overlapping the subproblems:

A problem can be broken down into subproblems which are reused several times.

(or)

A recursive algorithm for the problem solves the same sub problem over and over rather than always generating the new sub problems.

principal of optimality:-

A problem has optimal substructure if an optimal solution can be constructed efficiently from optimal solu-tions of sub problems.

(or)

we can solve larger sub problems

To overcome the drawback of the overlapping subproblem we use the following techniques.

1) Memorization
2) Tabularization

* In both memorization and tabularization we store the values of sub problems in a table.

* The difference between both of them is the way in which we store the values in the table.

Memorization:

In this technique the values of the subproblems are stored in the table using the following techniques.

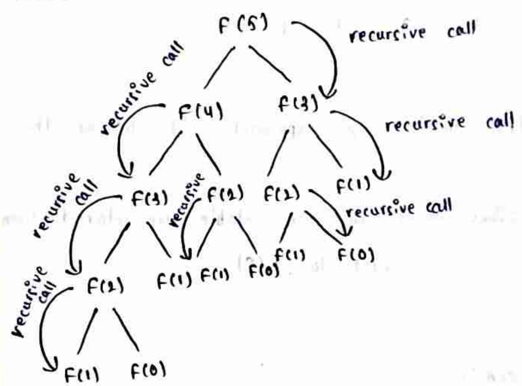
a) Top-down approach
b) Recursive method

Ex: we want to find out the f(5) in the fibonacci series, here $f(0)=0$, $f(1)=1$ and $f(n) = f(n-1) + f(n-2)$

Algorithm:

```
Algorithm fib(n)
{
  if (n<=1)
    return n;
  else
    return f(n-1) + f(n-2);
}
```

Tracing: To find f(5) so n=5



First we take the table with nill values

Nill	Nill	Nill	Nill	Nill	Nill
0	1	2	3	4	5

To compute the F(5) we will check the F(5) value in the table first, Intially we don't have anything in the table so we have to compute f(4) and f(3).

To compute f(4) and f(3) we have to check the table the table is empty, so compute f(2) and f(1) and to calculate f(2) we again need f(1) and f(0)

we have the values of f(1) and f(0) as '1' and '0' respectively so fill the values into the table.

0	1	2	3	4	5
0	1				

As we have f(0) and f(1) in table. compute the f(2) value
f(2) = f(1) + f(0) => 1. [Here f(1) and f(0) aren't computed, just taken from table]

0	1	2	3	4	5
0	1	1			

Like wise we calculate the values by taking the sub values from the table without again computing them.

f(3) = f(2) + f(1) = 2

f(4) = f(3) + f(2) = 3

f(5) = f(4) + f(3) = 5

0	1	2	3	4	5
0	1	1	2	3	5

Tabularization:

It follows the bottom-up approach, it follows the iteration method.

* Here the ~~functions~~ values in the table are stored from bottom to top i.e., f(0) to f(5)

Algorithm:

```
algorithm iteration_feb (n)
{
  if (n<=1)
    return n;
  else
  {
    f(0) = 0 ;
    f(1) = 1 ;
    for ( i=2 ; i<=n; i++)
}   f(i) = f(i-1) + f(i-2) ;
    return f(n);
}
```

Divide & Conquer (vs) Dynamic programming:

* In both the techniques the problem is divided into sub problems, in dynamic programming the same sub problem is repeatedly called with the same values. where as in Divide & Conquer the sub problems are called repeatedly, but not the same sub problems.

Dynamic programming (vs) Greedy method: Both are used to solve the optimization problems.

* In the greedy method decision is taken one time and we will follow that decision where as in dynamic programming we try to find out all possible solutions and then we pick up the best solution, dynamic programming is time consuming compared to greedy method.

All pairs shortest path problem:

Here a weighted directed graph is given, we have to find out the cost from / distance from every node to every other node in the graph. Here all the nodes are sources and all are the destinations also.

* Here the cost from vertex to itself is considered as zero.

* Here in order to solve the all pairs shortest-path problem ~~floyd~~ warshell algorithm is used.

* If we use the Dijkstra's algorithm to solve this problem, then the time complexity $O(v^3 \log v)$ where 'v' is the no.of vertices, to reduce the time complexity we don't use the Dijkstra's algorithm for all-pair shortest path problem.

Note:

for Dijkstra's algorithm time complexity $O(E \log v)$, if the graph is dense graph $E = v^2$ then $O(E \log v) = O(v^2 \log v)$ for one vertex the time complexity is $O(v^2 \log v)$ and for all vertices $O(V \cdot v^2 \log v) = O(v^3 \log v)$ which is greater than the floyd warshall algorithm

09/09/19

* In order to find out the shortest path from every node to every other node, construct the adjacency matrices $D^0, D^1, D^2 \ldots D^n$ where 'n' is the no. of vertices.

- The adjacency matrix D^0 having the direct path cost from one vertex to another vertex.
- The adjacency matrix D^1 having the cost from one vertex to another vertex via vertex 1
- Likewise D^2 is having the cost from one vertex to another vertex via vertex 2.

  Similarly D^n is having the cost from one vertex to another vertex via vertex 'n'.

* D^n contains the / having the ~~cost~~ ~~from~~ ~~one~~ shortest distance from one vertex to another vertex.

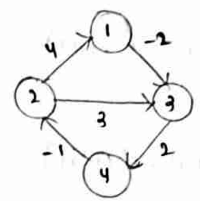
* The floyd warshall algorithms also works on -ve edge weights. (having no cycle with -ve weights)

Algorithm :

```
Algorithm all-pair ( w, D)
{
 for i=1 to n do
 {
  for j=1 to n do
  {
   D[i,j] = w[i,j];
  }
 }
 for k=1 to n do
 {
  for i=1 to n do
  {
   for j=1 to n do
   {
    D[i,j] = min ( D[i,j], D[i,k] + D[k,j] );
   }
  }
 }
}
```

* Time complexity is $O(n^3)$

Ex: Construct the shortest path from each vertex to every other vertex.



Ans: Step 1 : construct D^0 (the direct path from one vertex to another)

D^0	1	2	3	4
1	0	∞	-2	∞
2	4	0	3	∞
3	∞	∞	0	2
4	∞	-1	∞	0

* $D^k(i,j) = \min\left(D^{k-1}(i,j),\ D^{k-1}(i,k) + D^{k-1}(k,j) \right);$

Step 2 : D^1 Now the cost from one vertex to other vertices via vertex '1'.

$D^1(1,1) = \min\left(D^0(1,1),\ D^0(1,1) + D^0(1,1)\right)$

$D^1(1,1) = \min(0, 0) = 0.$

D^1	1	2	3	4
1	0	∞	-2	∞
2	4	0	2	∞
3	∞	∞	0	2
4	∞	-1	∞	0

* Here we are travelling from one vertex to other via vertex '1', so for the first Now we are travelling from vertex '1' to other vertices via '1' means it is same as travelling from vertex '1' to other vertices.

$D^1(2,1) = \min\left(D^0(2,1),\ D^0(2,1) + D^0(1,1)\right)$

$= \min(4, 4) = 4$

$D^1(2,3) = \min\{D^0(2,3),\ D^0(2,1) + D^0(1,3)\}$

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

$D^1(2,4) = \min\{D^0(2,4),\ D^0(2,1) + D^0(1,4)\}$

$= \min\{\infty,\ 4 + \infty\} = \infty$

$D^1(3,1) = \min\{D^0(3,1), D^0(3,1) + D^0(1,1)\}$

$= \min\{\infty, \infty + 0\} = \infty$

$D^1(3,2) = \min\{D^0(3,2), D^0(3,1) + D^0(1,2)\}$

$= \min\{\infty, \infty + \infty\} = \infty$

$D^1(3,4) = \min\{D^0(3,4), D^0(3,1) + D^0(1,4)\}$

$= \min\{2, \infty + \infty\} = 2$

$D^1(4,1) = \min\{D^0(4,1), D^0(4,1) + D^0(1,1)\}$

$= \min\{\infty, \infty + 0\} = \infty.$

$D^1(4,2) = \min\{-1, \infty + \infty\} = -1$

$D^1(4,3) = \min\{\infty, \infty + -2\} = \infty$

step 3 : D^2, means from one vertex to other vertices via vertex 2.

D^2	1	2	3	4
1	0	∞	-2	∞
2	4	0	2	∞
3	∞	∞	0	2
4	3	-1	1	0

$D^2(1,2) = \min\{D^1(1,2), D^1(1,2) + D^1(2,2)\}$

$D^2(1,2) = \min\{\infty, \infty + 0\} = \infty$

$D^2(1,3) = \min\{D^1(1,3), D^1(1,2) + D^1(2,3)\}$

$= \min\{-2, \infty + 3\} = -2$

$D^2(1,4) = \min\{D^1(1,4), D^1(1,2) + D^1(2,4)\}$

$= \min\{\infty, \infty + \infty\} = \infty$

$D^2(2,1) = \min\{D^1(2,1), D^1(2,2) + D^1(2,1)\}$

$= \min\{4, 0 + 4\} = 4.$

$D^2(2,3) = \min\{D^1(2,3), D^1(2,2) + D^1(2,3)\}$

$= \min\{2, 0 + 2\} = 2.$

$D^2(2,4) = \min\{D^1(2,4), D^1(2,2) + D^1(2,4)\}$

$= \min\{\infty, 0 + \infty\} = \infty.$

$D^2(3,1) = \min\{D^1(3,1), D^1(3,2) + D^1(2,1)\}$

$= \min\{\infty, \infty + 4\} = \infty$

$D^2(3,2) = \min\{D^1(3,2), D^1(3,2) + D^1(2,2)\}$

$= \min\{\infty, \infty + 0\} = \infty$

$D^2(3,4) = \min\{D^1(3,4), D^1(3,2) + D^1(2,4)\}$

$= \min\{2, \infty + \infty\} = 2$

$D^2(4,1) = \min\{D^1(4,1), D^1(4,2) + D^1(2,1)\}$

$= \min\{\infty, -1 + 4\} = 3.$

$D^2(4,2) = \min\{D^1(4,2), D^1(4,2) + D^1(2,2)\}$

$= \min\{-1, -1 + 0\} = -1$

$D^2(4,3) = \min\{D^1(4,3), D^1(4,2) + D^1(2,3)\}$

$= \min\{\infty, -1 + 2\} = 1$

step 3 : D^3, means cost from one vertex other via vertex 3.

D^3	1	2	3	4
1	0	∞	-2	0
2	4	0	2	4
3	∞	∞	0	2
4	3	-1	1	0

$D^3(1,2) = \min\{D^2(1,2), D^2(1,3) + D^2(3,2)\}$

$= \min\{\infty, -2 + \infty\} = \infty$

$D^3(1,3) = \min\{D^2(1,3), D^2(1,3) + D^2(3,3)\}$

$= \min\{-2, -2 + 0\} = -2$

$D^3(1,4) = \min\{D^2(1,4), D^2(1,3) + D^2(3,4)\}$

$= \min\{\infty, -2 + 2\} = 0.$

$D^3(2,1) = \min\{D^2(2,1), D^2(2,3) + D^2(3,1)\}$

$= \min\{4, 2 + \infty\} = 4$

$D^3(2,3) = \min\{D^2(2,3), D^2(2,3) + D^2(3,3)\}$

$= \min\{2, 2 + 0\} = 2$

$D^3(2,4) = \min\{D^2(2,4), D^2(2,3) + D^2(3,4)\}$

$= \min\{\infty, 2 + 2\} = 4$

$D^3(3,1) = \min\{D^2(3,1), D^2(3,3) + D^2(3,1)\}$

$= \min\{\infty, 0 + \infty\} = \infty$

$D^3(3,2) = \min\{D^2(3,2), D^2(3,3) + D^2(3,2)\}$

$= \min\{\infty, 0 + \infty\} = \infty$

$D^3(3,4) = \min\{D^2(3,4), D^2(3,3) + D^2(3,4)\}$

$= \min\{2, 0 + 2\} = 2$

$D^3(4,1) = \min\{D^2(4,1), D^2(4,3) + D^2(3,1)\}$

$= \min\{3, 1 + \infty\} = 3$

$D^3(4,2) = \min\{D^2(4,2),\ D^2(4,3) + D^2(3,2)\}$

$= \min\{-1,\ 1+\infty\} = -1$

$D^3(4,3) = \min\{D^2(4,3),\ D^2(4,3) + D^2(3,3)\}$

$= \min\{1,\ 1+0\} = 1.$

D^4 step 4 : D^4 means the cost from one vertex to other via vertex '4'.

$D^4(4,1) = \min\{D^3(4,1),\ D^3(4,3) + D^3(3,1)\}$

$= \min\{3,\ 1+\infty\} = 3.$

D^4	1	2	3	4
1	0	-1	-2	0
2	4	0	2	4
3	5	1	0	2
4	3	-1	1	0

$D^4(1,2) = \min\{D^3(1,2),\ D^3(1,4) + D^3(4,2)\}$

$= \min\{\infty,\ 0+(-1)\} = -1$

$D^4(1,3) = \min\{D^3(1,3),\ D^3(1,4) + D^3(4,3)\}$

$= \min\{-2,\ 0+1\} = -2$

$D^4(1,4) = \min\{D^3(1,4),\ D^3(1,4) + D^3(4,4)\}$

$= \min\{0,\ 0+0\} = 0.$

$D^4(2,1) = \min\{D^3(2,1),\ D^3(2,4) + D^3(4,1)\}$

$= \min\{4,\ 4+3\} = 4$

$D^4(2,3) = \min\{D^3(2,3),\ D^3(2,4) + D^3(4,3)\}$

$= \min\{2,\ 4+1\} = 2.$

$D^4(2,4) = \min\{D^3(2,4),\ D^3(2,4) + D^3(4,4)\}$

$= \min\{4,\ 4+0\} = 4.$

$D^4(3,1) = \min\{D^3(3,1),\ D^3(3,4) + D^3(4,1)\}$

$= \min\{\infty,\ 2+3\} = 5$

$D^4(3,2) = \min\{D^3(3,2),\ D^3(3,4) + D^3(4,2)\}$

$= \min\{\infty,\ 2+-1\} = 1.$

$D^4(3,4) = \min\{D^3(3,4),\ D^3(3,4) + D^3(4,4)\}$

$= \min\{2,\ 2+0\} = 2.$

$D^4(4,2) = \min\{D^3(4,2),\ D^3(4,4) + D^3(4,2)\}$

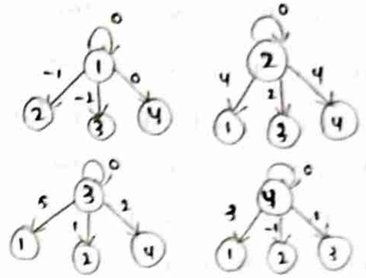
$= \min\{-1,\ 0+(-1)\} = -1$

$D^4(4,3) = \min\{D^3(4,3),\ D^3(4,4) + D^3(4,3)\}$

$= \min\{1,\ 0+1\} = 1$

$\therefore$ The final minimum cost paths from one vertex to every other vertices are

	1	2	3	4
1	0	-1	-2	0
2	4	0	2	4
3	5	1	0	2
4	3	-1	1	0



Note :

while we are constructing the matrices $D^0, D^1, D^2, \ldots, D^n$

* The distance from node to itself will be zero.

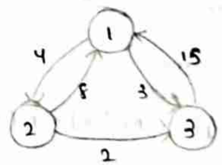
* If we are constructing

D^1 – the first row and first column will be constant

D^2 – the second row and second column will be constant

D^n – the n^{th} row and n^{th} column will be constant with respect to the $D^0, D^1, \ldots, D^{n-1}$ respectively so there is no need to check there values

Algorithm tracing:



Algorithm all_pair (w, D)

{

for i:=1 to 3 do → true

{

for j:=1 to 3 do → true

{ D[1, 1] = w[1, 1] = 0;

- for j=2 to 3 do → true

  D[1,2] = w[1,2] ; ⇒ D[1,2] = 4

- for j=3 to 3 do → true

  D[1,3] = w[1,3] ; ⇒ D[1,3] = 3

  - for j=4 to 3 do → false

→ for i=2 to 3 do → true

- for j=1 to 3 do → true

  D[2,1] = w[2,1] ⇒ D[2,1] = 8

- for j=2 to 3 do → true

  D[2,2] = w[2,2] ⇒ D[2,2] = 0

- for j=3 to 3 do → true

  D[2,3] = w[2,3] ⇒ D[2,3] = 2

- for j=4 to 3 do → false

D^0	1	2	3
1	0	4	3
2	8	0	2
3	15	∞	0

→ for i=3 to 3 do → true

- for j=1 to 3 do → true

  D[3,1] = w[3,1] ⇒ D[3,1] = 15

- for j=2 to 3 do → true

  D[3,2] = w[3,2] ⇒ D[3,2] = ∞

- for j=3 to 3 do → true

  D[3,3] = w[3,3] ⇒ D[3,3] = 0

- for j=4 to 3 do → false.

→ for i=4 to 3 do → false.

D^1	1	2	3
1	0	4	3
2	8	0	2
3	15	19	0

- \> for k=1 to 3 do → true

  → for i=1 to 3 do → true

  - for j=1 to 3 do → true

    D[i,j] = min (D[i,j] , D[i,k] + D[k,j]) ;

    ⇒ D[1,1] = min (D[1,1] , D[1,1] + D[1,1]) ;

    D[1,1] = min (0, 0+0) = 0

  - for j=2 to 3 do → true

    D[1,2] = min (D[1,2], D[1,1] + D[1,2]) ;

    D[1,2] = min (4, 0+4) = 4 .

- for j=3 to 3 do → true

  D[1,3] = min (D[1,3] , D[1,1] + D[1,3]) ;

  D[1,3] = min (3, 0+3) = 3 .

- for j=4 to 3 do → false

→ for i=2 to 3 do → true

- for j=1 to 3 do → true

  D[2,1] = min (D[2,1] , D[2,1] + D[1,1]) ;

  = min (8, 8+0) = 8

- for j=2 to 3 do → true

  D[2,2] = min (D[2,2] , D[2,1] + D[1,2]) ;

  = min (0, 8+4) = 0 .

- for j=3 to 3 do → true

  D[2,3] = min (D[2,3] , D[2,1] + D[1,3]) ;

  = min (2 , 8+3) = 2 .

- for j=4 to 3 do → false

→ for i=3 to 3 do → true

- for j=1 to 3 do → true

  D[3,1] = min (D[3,1] , D[3,1] + D[1,1]) ;

  = min (15, 15+0) = 15

- for j=2 to 3 do → true

  D[3,2] = min (D[3,2] , D[3,1] + D[1,2]) ;

  = min (∞, 15+4) = 19

- for j=3 to 3 do → true

  D[3,3] = min (D[3,3], D[3,1] + D[1,3]) ;

  = min (0, 15+3) = 0

- for j=4 to 3 do → false .

→ for i=4 to 3 do → false .

D^2	1	2	3
1	0	4	3
2	8	0	2
3	15	19	.

- \> for k=2 to 3 do → true

  → for i=1 to 3 do → true

  - for j=1 to 3 do → true

    D[1,1] = min (D[1,1] , D[1,2] + D[2,1])

    = min (0, 4+8) = 0

- for $j:=2$ to 3 do → true

  $D[1,2] = \min(D[1,2],\ D[1,2] + D[2,2])$

  $= \min(4,\ 4+0) = 4.$

- for $j:=3$ to 3 do → true

  $D[1,3] = \min(D[1,3],\ D[1,2] + D[2,3])$

  $= \min(3,\ 4+2) = 3.$

→ for $i:=2$ to 3 do → true

- for $j:=1$ to 3 do → true

  $D[2,1] = \min(D[2,1],\ D[2,2] + D[2,1]);$

  $= \min(8,\ 0+8) = 8$

- for $j:=2$ to 3 do → true

  $D[2,2] = \min(D[2,2],\ D[2,2] + D[2,2]);$

  $= \min(0,\ 0+0) = 0$

- for $j:=3$ to 3 do → true.

  $D[2,3] = \min(D[2,3],\ D[2,2] + D[2,3]);$

  $= \min(2,\ 0+2) = 2.$

- for $j:=4$ to 3 do → false.

→ for $i:=3$ to 3 do → true

- for $j:=1$ to 3 do → true.

  $D[3,1] = \min(D[3,1],\ D[3,2] + D[2,1]);$

  $= \min(15,\ 19+8) = 15$

- for $j:=2$ to 3 do → true

  $D[3,2] = \min(D[3,2],\ D[3,2] + D[2,2]);$

  $= \min(19,\ 19+0) = 19$

- for $j:=3$ to 3 do → true

  $D[3,3] = \min(D[3,3],\ D[3,2] + D[2,3])$

  $= \min(0,\ 19+2) = 0.$

- for $j:=4$ to 3 do → false.

→ for $i:=4$ to 3 do → false

> for $k:=3$ to 3 do → true.

→ for $i:=1$ to 3 do → true

- for $j:=1$ to 3 do → true

  $D[1,1] = \min(D[1,1],\ D[1,3] + D[3,1]);$

  $= \min(0,\ 3+15) = 0.$

- for $j:=2$ to 3 do → true

  $D[1,2] = \min(D[1,2],\ D[1,3] + D[3,2]);$

  $= \min(4,\ 3+19) = 4.$

- for $j:=3$ to 3 do → true

  $D[1,3] = \min(D[1,3],\ D[1,3] + D[3,3])$

  $= \min(3,\ 3+0) = 3.$

- for $j:=4$ to 3 do → false.

→ for $i:=2$ to 3 do → true

- for $j:=1$ to 3 do → true.

  $D[2,1] = \min(D[2,1],\ D[2,3] + D[3,1]);$

  $= \min(8,\ 2+15) = 8$

- for $j:=2$ to 3 do → true

  $D[2,2] = \min(D[2,2],\ D[2,3] + D[3,2]);$

  $= \min(0,\ 2+19) = 0$

- for $j:=3$ to 3 do → true

  $D[2,3] = \min(D[2,3],\ D[2,3] + D[3,3]);$

  $= \min(2,\ 2+0) = 2.$

- for $j:=4$ to 3 do → false

→ for $i:=3$ to 3 do → true

- for $j:=1$ to 3 do → true

  $D[3,1] = \min(D[3,1],\ D[3,3] + D[3,1]);$

  $= \min(15,\ 0+15) = 15$

- for $j:=2$ to 3 do → true

  $D[3,2] = \min(D[3,2],\ D[3,3] + D[3,2]);$

  $= \min(19,\ 0+19) = 19$

- for $j:=3$ to 3 do → true

  $D[3,3] = \min(D[3,3],\ D[3,3] + D[3,3]);$

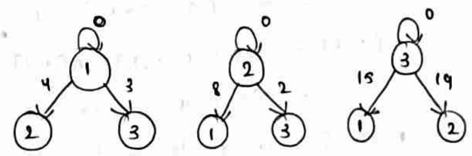
$= \min(0, 0+0) = 0$

- for $j=4$ to 3 do $\rightarrow$ false

$\rightarrow$ for $i=4$ to 3 do $\rightarrow$ false

$>$ for $k=4$ to 3 do $\rightarrow$ false

D	1	2	3
1	0	4	3
2	8	0	2
3	15	19	0



Time complexity :

The time complexity of all-pairs shortest path problem algorithm is $O(n^3)$

Matrix chain multiplication :

* 'n' matrices are given we have to find out the order in which the matrices are to be multiplied to reduce the number of scalar multiplications.
* The number of scalar multiplications is treated as the cost of multiplying the matrices.
* Scalar multiplication is nothing but element to element multiplication operation.
* Let us take two matrices of order 2x2 and 2x2 as follows

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$\begin{bmatrix} ae+bg & af+bh \\ ce+dg & ef+dh \end{bmatrix}$$

The resultant matrix order is 2x2, the number of elements in the resultant matrix are $2 \times 2 = 4$, to get each element in the resultant matrix, the number of multiplications required is 2

So the total number of element to element multiplication operations are

$$\underbrace{2 \times 2}_{\text{order}} \times \underbrace{2}_{\substack{\text{no.of multiplication} \\ \text{to get an element}}} = 8$$

* Let us consider two matrices $A_{1\times2}$, $B_{2\times3}$, $C_{3\times4}$
* Let us find the order in which the matrices are multiplied to reduce the cost.
* There are two ways to multiply the above matrices.

1) $(A_{1\times2} \, B_{2\times3}) \, C_{3\times4}$ 2) $A_{1\times2} \, (B_{2\times3} \, C_{3\times4})$

1st method :

$(A\ B)\ C$

$\underbrace{(A\ B)_{1\times 3}}_{1\times 3\times 2}\ C_{3\times 4}$

$(A\ B\ C)_{1\times 4\times 3}$

$\text{cost} = (1\times 4\times 3) + (1\times 3\times 2) = 18$

2nd method :

$A\ (B\ C)$

$A_{1\times 2}\ \underbrace{(B\ C)_{2\times 4}}_{2\times 3\times 4}$

$(A\ B\ C)_{1\times 4}$

$\text{cost} = (1\times 4\times 2) + (2\times 3\times 4) = 8+24 = 32$

In the ~~second~~ first method, the number of scalar multiplications are 18, so we follow the 1st method to multiply 3 given matrices.

* Suppose if there are 4 matrices and we have to find out the multiplication of the 4 matrices, then it is difficult to list out all the ways and finding the cost in each way.

* To solve this problem using dynamic programming method we will perform the following steps.

step 1 : Let M_{ij} denotes the cost of multiplying A_i to A_j, here $m_{ii} = 0\ \forall\ i$

step 2 : $M[1,n]$ is the required solution

step 3 : Here sequence of decisions can be built using the principal of optimality. we will apply the following formula for computing each sequence.

$$M_{ij} = \min\{m_{ik} + m_{k+1,j} + P_i P_{k+1} P_{j+1},\ i \le k < j\}$$

12/09/14

Example :

Find the minimum cost of multiplying the matrices with order $A_1 = 1\times 3$, $A_2 = 3\times 2$, $A_3 = 2\times 4$, $A_4 = 4\times 6$.

Ans: Given A_1 order is 1×3 $(P_1 \times P_2)$

A_2 order is 3×2 $(P_2 \times P_3)$

A_3 order is 2×4 $(P_3 \times P_4)$

A_4 order is 4×6 $(P_4 \times P_5)$

$P_1 = 1$, $P_2 = 3$, $P_3 = 2$, $P_4 = 4$, $P_5 = 6$.

$$M_{ij} = \min\{m_{ik} + m_{k+1,j} + P_i P_{k+1} P_{j+1},\ i \le k < j\}$$

1	2	3	4
$K=1$ $M_{11}=0$	$K=2$ $M_{22}=0$	$K=3$ $M_{33}=0$	$K=4$ $M_{44}=0$
$M_{12}=6$ $K=1$	$M_{23}=24$ $K=2$	$M_{34}=48$ $K=3$	
$M_{13}=14$ $K=2$	$M_{24}=84$ $K=2$		
$M_{14}=38$ $K=3$			

All values which are in the form of $M_{ii} = 0$ for every i (i.e., $M_{11} = M_{22} = M_{33} = M_{44} = 0$)

M_{12}, Here $i=1$ and $j=2$ $\Rightarrow$ $i \le k < j$, means $K=1$.

- $M_{12} = \min\{M_{11} + M_{22} + P_1 P_2 P_3\}$
  $K=1$ $= \min\{0+0+1\times 3\times 2\} = 6$.
- $M_{23} = \min\{M_{22} + M_{33} + P_2 P_3 P_4\}$
  $K=2$ $= 0+0+3\times 2\times 4 = 24$.
- $M_{34} = \min\{M_{33} + M_{44} + P_3 P_4 P_5\}$
  $K=3$ $= 0+0+2\times 4\times 6 = 48$

- $M_{13} = \min\{ M_{11} + M_{23} + P_1 P_2 P_4 ,\ M_{12} + M_{33} + P_1 P_3 P_4 \}$

$K = 1,2 = \min\{ 0 + 24 + 1\times3\times4 ,\ 6 + 0 + 1\times2\times4 \}$

$= \min\{ 36, 14 \} = 14.$

- $M_{24} = \min\{ M_{22} + M_{34} + P_2 P_3 P_5 ,\ M_{23} + M_{44} + P_2 P_4 P_5 \}$

$K = 2,3 = \min\{ 0 + 48 + 3\times2\times6 ,\ 24 + 0 + 3\times4\times6 \}$

$= \min\{ 84, 96 \} = 84.$

- $M_{14} = \min\{ M_{11} + M_{24} + P_1 P_2 P_5 ,\ M_{12} + M_{34} + P_1 P_3 P_5 ,$

$K = 1,2,3 \qquad M_{13} + M_{44} + P_1 P_4 P_5 \}$

$= \min\{ 0 + 84 + 1\times3\times6 ,\ 6 + 48 + 1\times2\times6 ,$

$14 + 0 + 1\times4\times6 \}$

$= \min\{ 102, 66, 38 \} = 38.$

Here the final operation is M_{14} we got the minimum cost at $K_3 = 3$.

1.) $A_1 \ A_2 \ A_3 \ A_4$

$\downarrow$

split at $K=3$.

$\rightarrow (A_1 \ A_2 \ A_3) \ A_4$

M_{13}, we got minimum cost at $K=2$,

split at $K=2$.

$\rightarrow ((A_1 \ A_2) \ A_3) \ A_4$

$\downarrow$

M_{12} we got minimum cost at $K=1$, split at $K=1$

$\rightarrow ((((A_1) \ A_2) \ A_3) \ A_4)$

1st

2nd

3rd

$(A_1 A_2) \Rightarrow (A_1 A_2) A_3 \Rightarrow (A_1 A_2 A_3) A_4$

$1\times3 \quad 3\times2 \qquad 1\times2 \quad 2\times4 \qquad 1\times4 \quad 4\times6$

$1\times2\times3 = 6 \qquad 1\times4\times2 = 8 \qquad 1\times6\times4 = 24$

$\therefore$ Total cost $= 6 + 8 + 24 = 38$

Ex 2: $A_1 = 5\times4$, $A_2 = 4\times6$, $A_3 = 6\times2$, $A_4 = 2\times7$.

Ans: Given, A_1 order is 5×4 $(P_1 \times P_2)$

A_2 order is 4×6 $(P_2 \times P_3)$

A_3 order is 6×2 $(P_3 \times P_4)$

A_4 order is 2×7 $(P_4 \times P_5)$

Here $P_1 = 5$, $P_2 = 4$, $P_3 = 6$, $P_4 = 2$, $P_5 = 7$.

1	2	3	4
$M_{11}=0$	$M_{22}=0$	$M_{33}=0$	$M_{44}=0$
$M_{12}=120$ $K=1$	$M_{23}=48$ $K=2$	$M_{34}=84$ $K=3$	
$M_{13}=88$ $K=1$	$M_{24}=104$ $K=3$		
$M_{14}=158$ $K=3$			

$M_{ij} = \min\{ m_{ik} + m_{k+1,j} + P_i P_{k+1} P_{j+1} ,\ i \le k < j \}$

- $M_{12} = \min\{ m_{11} + M_{22} + P_1 P_2 P_3 \}$

$K=1 = \min\{ 0 + 0 + 5\times4\times6 \} = 120.$

- $M_{23} = \min\{ m_{22} + M_{33} + P_2 P_3 P_4 \}$

$K=2 \quad = 0 + 0 + 4\times6\times2 = 48$

- $M_{34} = \min\{ m_{33} + M_{44} + P_3 P_4 P_5 \}$

$K=3 \quad = 0 + 0 + 6\times2\times7 = 84$

- $M_{13} = \min\{ M_{11} + m_{23} + P_1 P_2 P_4 ,\ m_{12} + m_{33} + P_1 P_3 P_4 \}$

$K = 1,2 = \min\{ 0 + 48 + 5\times4\times2 ,\ 120 + 0 + 5\times6\times2 \}$

$= \min\{ 88, 180 \} = 88$

- $M_{24} = \min\{ m_{23} + m_{34} + P_2 P_3 P_5,\ m_{23} + m_{44} + P_2 P_4 P_5 \}$

$k=2,3$ $= \min\{ 0 + 84 + 4\times 6\times 7,\ 48 + 0 + 4\times 2\times 7 \}$

$= \min\{ 252,\ 104 \} = 104$

- $M_{14} = \min\{ m_{11} + m_{24} + P_1 P_2 P_5,\ m_{12} + m_{34} + P_1 P_3 P_5,$

$k=1,2,3$ $m_{13} + m_{44} + P_1 P_4 P_5 \}$

$= \min\{ 0 + 104 + 5\times 4\times 7,\ 120 + 84 + 5\times 6\times 7,$

$88 + 0 + 5\times 2\times 7 \}$

$= \min\{ 244,\ 414,\ 158 \} = 158$

The final operation is M_{14} we got minimum cost at $k=3$, so split there.

$(A_1\ A_2\ A_3)\ A_4$

$\downarrow$

M_{13}, the minimum cost at $k=1$, split there

$((A_1)(A_2 A_3)\ A_4)$

$\downarrow$

M_{23}, minimum cost at $k=2$, split there

$((A_1)((A_2)\ A_3),\ A_4)$

$A_2\ A_3 \Rightarrow (A_1)\ (A_2 A_3) \Rightarrow (A_1\ A_2 A_3),\ A_4$

$4\times 6 \quad 6\times 2 \qquad 5\times 4 \quad 4\times 2 \qquad 5\times 2 \quad 2\times 7$

$4\times 2\times 6 = 48 \qquad 5\times 2\times 4 = 40 \qquad 5\times 7\times 2 = 70$

$\therefore$ Total cost is $= 48 + 40 + 70$

$= 158$

Ex 3: $A_1 = 10\times 5$, $A_2 = 5\times 20$, $A_3 = 20\times 30$, $A_4 = 30\times 6$.

Ans: Given A_1 order is 10×5 $(P_1\times P_2)$

A_2 order is 5×20 $(P_2\times P_3)$

A_3 order is 20×30 $(P_3\times P_4)$

A_4 order is 30×6 $(P_4\times P_5)$

Here $P_1 = 10$, $P_2 = 5$, $P_3 = 20$, $P_4 = 30$, and $P_5 = 6$.

1	2	3	4
$M_{11}=0$	$M_{22}=0$	$M_{33}=0$	$M_{44}=0$
$M_{12}=1000$ $k=1$	$M_{23}=3000$ $k=2$	$M_{34}=3600$ $k=3$	
$M_{13}=4500$ $k=1$	$M_{24}=3900$ $k=3$		
$M_{14}=4200$ $k=1$			

$\rightarrow M_{ij} = \min\{ m_{ik} + m_{k+1,j} + P_i P_{k+1} P_{j+1},\ i\le k<j \}$

- $M_{12} = \min\{ m_{11} + m_{22} + P_1 P_2 P_3 \}$

$k=1$ $= \min\{ 0 + 0 + 10\times 5\times 20 \} = 1000$

- $M_{23} = \min\{ m_{22} + M_{33} + P_2 P_3 P_4 \}$

$k=2$ $= \min\{ 0 + 0 + 5\times 20\times 30 \} = 3000$

- $M_{34} = \min\{ m_{33} + m_{44} + P_3 P_4 P_5 \}$

$k=3$ $= \min\{ 0 + 0 + 20\times 30\times 6 \} = 3600$

- $M_{13} = \min\{ m_{11} + m_{23} + P_1 P_2 P_4,\ m_{12} + m_{33} + P_1 P_3 P_4 \}$

$k=1,2$ $= \min\{ 0 + 3000 + 10\times 5\times 30,\ 1000 + 0 + 10\times 20\times 30 \}$

$= \min\{ 4500,\ 7000 \} = 4500$

- $M_{24} = \min\{ m_{22} + M_{34} + P_2 P_3 P_5,\ m_{23} + m_{44} + P_2 P_4 P_5 \}$

$k=2,3$ $= \min\{ 0 + 3600 + 5\times 20\times 6,\ 3000 + 0 + 5\times 30\times 6 \}$

$= \min\{ 4200,\ 3900 \} = 3900$

- $M_{14} = \min\{ m_{11} + m_{24} + P_1 P_2 P_5,\ m_{12} + m_{34} + P_1 P_3 P_5,$

$k=1,2,3$ $m_{13} + m_{44} + P_1 P_4 P_5 \}$

$M_{14} = \min\{ 0 + 3900 + 10\times5\times6,\ 1000 + 3600 + 10\times20\times6,$

$4500 + 0 + 10\times30\times6 \}$

$= \min\{4200, 5800, 6300\} = 4200$

The final operation is M_{14}, is minimum cost at $k=1$ split there.

$(A_1)\ A_2\ A_3\ A_4$

$\downarrow$

M_{24}, minimum cost at $k=2$.

$(A_1)\ (A_2\ A_3)\ A_4$

$\downarrow$

M_{23}, minimum cost at $k=2$.

$((A_1)\ ((A_2)\ A_3)\ A_4)$

$A_2\ A_3 \Rightarrow (A_2\ A_3)\ A_4 \Rightarrow (A_1)\ (A_2 A_3 A_4)$

$\underbrace{5\times20 \quad 20\times30}_{5\times30\times20 = 3000}$ $\quad \underbrace{5\times30 \quad 30\times6}_{5\times6\times30 = 900}$ $\quad \underbrace{10\times5 \quad 5\times6}_{10\times6\times5 = 300}$

$\therefore$ Total cost $= 3000 + 900 + 300$
$= 4200$

13/09/19

Algorithm matrix-chain-mul( )

```
{
  for i:=1 to n do
      m[i,i] = 0;
  for len=2 to n do
  {
    for i:=1 to (n-len+1) do
    {
      j = i+len-1;
      M[i,j] = ∞;
      for k=i to j-1 do
      {
        q = m[i,k] + m[k+1, j] + p[i-1] * p[k] * p[j];
        if (q < m[i,j])
        {
          M[i,j] = q;
          s[i,j] = k;
        }
      }
    }
  }
  return M[1,n];
}
```

Time complexity :

The algorithm time complexity is $O(n^3)$

17-09/19

Travelling Sales person problem :

A sales man is there and cities are there the sales man travelled cities to sell the items. The problem here is the sales man starts at a city where he is living and then he travel to the remaining cities in the country. Only once and then he come back to his place, where he is started his journey. There will be several paths to cover all the cities and going back to the city where he is started. Our goal is to find the minimum cost path.

* Here a weighted directed graph will be given, the nodes in the cities are assumed as the cities and the edges in the graph are assumed as the paths (or) routes to travel from one city to the other city.

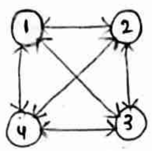
* The weights associated with the edges are taken as the distance (or) travel time (or) cost to move from one city to the other city.

* Here, a cycle is formed because of the sales man comback to the city where he started

after visiting all the cities in the country.

18-09-19

problem :



$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

source = 1

Ans: formula = $g(i,s) = \min \{ d(i,j) + g(j, \{s - \{j\}\}) \}, j \in s$

$g(1, \{2,3,4\}) =$

$$\min \begin{cases} d(1,2) = 10 + g(2, \{3,4\}) = 25 \Rightarrow 35 \\ d(1,3) = 15 + g(3, \{2,4\}) = 25 \Rightarrow 40 \\ d(1,4) = 20 + g(4, \{2,3\}) = 23 \Rightarrow 43 \end{cases}$$

$\rightarrow g(2, \{3,4\}) = \min \begin{cases} d(2,3) = 9 + g(3, \{4\}) = 20 \Rightarrow 29 \\ d(2,4) = 10 + g(4, \{3\}) = 15 \Rightarrow 25 \end{cases}$

$g(3, \{4\}) = d(3,4) + g\{4, \emptyset\} = 12 + 8 = 20$

$g\{4, \emptyset\} = d(4,1) = 8$

$g(4, \{3\}) = d(4,3) + g\{3, \emptyset\} = 9 + 6 = 15$

$g(3, \emptyset) = d(3,1) = 6$

$\rightarrow g(3, \{2,4\}) = \min \begin{cases} d(3,2) = 13 + g(2, \{4\}) = 18 \Rightarrow 31 \\ d(3,4) = 12 + g(4, \{2\}) = 13 \Rightarrow 25 \end{cases}$

$g(2, \{4\}) = d(2,4) + g\{4, \emptyset\} = 10 + 8 = 18$

$g\{4, \emptyset\} = d(4,1) = 8$

$g(4, \{2\}) = d(4,2) + g\{2, \emptyset\} = 8 + 5 = 13$

$g\{2, \emptyset\} = d(2,1) = 5$

$\rightarrow g(4, \{2,3\}) = \min \begin{cases} d(4,2) = 8 + g(2, \{3\}) = 15 \Rightarrow 23 \\ d(4,3) = 9 + g(3, \{2\}) = 18 \Rightarrow 27 \end{cases}$

$g(2, \{3\}) = d(2,3) + g(3, \emptyset) = 9 + 6 = 15$

$g(3, \emptyset) = d(3,1) = 6$

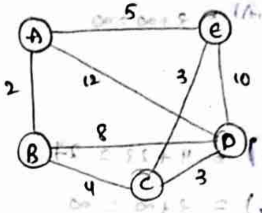
$g(3, \{2\}) = d(3,2) + g\{2, \emptyset\} = 13 + 5 = 18$

$g(2, \emptyset) = d(2,1) = 5$

$\therefore$ Distance is (path) 1 - 2 - 4 - 3 - 1

$\therefore g\{1, \{2,3,4\}\} = 35$

2)



$$\begin{array}{c|c} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{bmatrix} 0 & 2 & \infty & 12 & 5 \\ 2 & 0 & 4 & 8 & \infty \\ \infty & 4 & 0 & 3 & 3 \\ 12 & 8 & 3 & 0 & 10 \\ 5 & \infty & 3 & 10 & 0 \end{bmatrix} \end{array}$$

source = A

Ans: $g(A, \{B,C,D,E\}) =$

$$\min \begin{cases} d(A,B) + g(B, \{C,D,E\}) = 2 + 19 = 21 \\ d(A,C) + g(C, \{B,D,E\}) = \infty + 23 = \infty \\ d(A,D) + g(D, \{B,C,E\}) = 12 + 19 = 31 \\ d(A,E) + g(E, \{B,C,D\}) = 5 + 16 = 21 \end{cases}$$

A - B - D - C - E - A

a) $g(B, \{C,D,E\})$

$$\min \begin{cases} d(B,C) + g(C, \{D,E\}) \Rightarrow 4 + 18 = 22 \\ d(B,D) + g(D, \{C,E\}) \Rightarrow 8 + 11 = 19 \\ d(B,E) + g(E, \{C,D\}) \Rightarrow \infty + 18 = \infty \end{cases}$$

i) $g(C, \{D,E\}) = \min \begin{cases} d(C,D) + g(D, \{E\}) = 15 \Rightarrow 18 \\ d(C,E) + g(E, \{D\}) = 22 \Rightarrow 25 \end{cases}$

$g(D, \{E\}) = d(D,E) + g\{E, \emptyset\} = 10 + 5 = 15$

$g\{E, \emptyset\} = d(E,A) = 5$

$g(E, \{D\}) = d(E,D) + g\{D, \emptyset\} = 10 + 12 = 22$

$g(D, \emptyset) = d(D,A) = 12$

ii) $g(D, \{C,E\}) = \min \begin{cases} d(D,C) + g(C, \{E\}) = 3 + 8 = 11 \\ d(D,E) + g(E, \{C\}) = \infty \end{cases}$

$g(C, \{E\}) = d(C,E) + d(E,A) = 3 + 5 = 8$

$g(E, \{C\}) = d(E,C) + d(C,A) = 3 + \infty = \infty$

iii) $g(E,\{C,D\}) = \min \begin{cases} d(E,C) + g(C,\{D\}) = 3+15 = 18 \\ d(E,D) + g(D,\{C\}) = 10+\infty = \infty \end{cases}$

$g(C,\{D\}) = d(C,D) + d(D,A) = 3+12 = 15$

$g(D,\{C\}) = d(D,C) + d(C,A) = 3+\infty = \infty$

b) $g(C,\{B,D,E\}) = \min \begin{cases} d(C,B) + g(B,\{D,E\}) = 4+23 = 27 \\ d(C,D) + g(D,\{B,E\}) = 3+\infty = \infty \\ d(C,E) + g(E,\{B,D\}) = 3+20 = 23 \end{cases}$

i) $g(B,\{D,E\}) = \min \begin{cases} d(B,D) + g(D,\{E\}) = 15+8 = 23 \\ d(B,E) + g(E,\{D\}) = 22+\infty = \infty \end{cases}$

$g(D,\{E\}) = d(D,E) + d(E,A) = 10+5 = 15$

$g(E,\{D\}) = d(E,D) + d(D,A) = 10+12 = 22$

ii) $g(D,\{B,E\}) = \min \begin{cases} d(D,B) + g(B,\{E\}) = 8+\infty = \infty \\ d(D,E) + g(E,\{B\}) = 10+\infty = \infty \end{cases}$

$g(B,\{E\}) = d(B,E) + d(E,A) = \infty+5 = \infty$

$g(E,\{B\}) = d(E,B) + d(B,A) = \infty+2 = \infty$

iii) $g(E,\{B,D\}) = \min \begin{cases} d(E,B) + g(B,\{D\}) = \infty+20 = \infty \\ d(E,D) + g(D,\{B\}) = 10+10 = 20 \end{cases}$

$g(B,\{D\}) = d(B,D) + d(D,A) = 8+12 = 20$

$g(D,\{B\}) = d(D,B) + d(B,A) = 8+2 = 10$

c) $g(D,\{B,C,E\}) = \min \begin{cases} d(D,B) + g(B,\{C,E\}) = 8+12 = 20 \\ d(D,C) + g(C,\{B,E\}) = 3+\infty = \infty \\ d(D,E) + g(E,\{B,C\}) = 10+9 = 19 \end{cases}$

i) $g(B,\{C,E\}) = \min \begin{cases} d(B,C) + g(C,\{E\}) = 4+8 = 12 \\ d(B,E) + g(E,\{C\}) = \infty \end{cases}$

$g(C,\{E\}) = d(C,E) + d(E,A) = 3+5 = 8$

$g(E,\{C\}) = d(E,C) + d(C,A) = 3+\infty = \infty$

ii) $g(C,\{B,E\}) = \min \begin{cases} d(C,B) + g(B,\{E\}) = 4+\infty = \infty \\ d(C,E) + g(E,\{B\}) = 3+\infty = \infty \end{cases}$

$g(B,\{E\}) = d(B,E) + d(E,A) = \infty+5 = \infty$

$g(E,\{B\}) = d(E,B) + d(B,A) = \infty+2 = \infty$

iii) $g(E,\{B,C\}) = \min \begin{cases} d(E,B) + g(B,\{C\}) = \infty+\infty = \infty \\ d(E,C) + g(C,\{B\}) = 3+6 = 9 \end{cases}$

$g(B,\{C\}) = d(B,C) + d(C,A) = 4+\infty = \infty$

$g(C,\{B\}) = d(C,B) + d(B,A) = 4+2 = 6$

d) $g(E,\{B,C,D\}) = \min \begin{cases} d(E,B) + g(B,\{C,D\}) = \infty+19 = \infty \\ d(E,C) + g(C,\{B,D\}) = 3+13 = 16 \\ d(E,D) + g(D,\{B,C\}) = 10+9 = 19 \end{cases}$

A-E-C-D-

i) $g(B,\{C,D\}) = \min \begin{cases} d(B,C) + g(C,\{D\}) = 4+15 = 19 \\ d(B,D) + g(D,\{C\}) = 8+\infty = \infty \end{cases}$

$g(C,\{D\}) = d(C,D) + d(D,A) = 3+12 = 15$

$g(D,\{C\}) = d(D,C) + d(C,A) = 3+\infty = \infty$

ii) $g(C,\{B,D\}) = \min \begin{cases} d(C,B) + g(B,\{D\}) = 4+20 = 24 \\ d(C,D) + g(D,\{B\}) = 3+10 = 13 \end{cases}$

$g(B,\{D\}) = d(B,D) + d(D,A) = 8+12 = 20$

$g(D,\{B\}) = d(D,B) + d(B,A) = 8+2 = 10$

iii) $g(D,\{B,C\}) = \min \begin{cases} d(D,B) + g(B,\{C\}) = 8+\infty = \infty \\ d(D,C) + g(C,\{B\}) = 3+6 = 9 \end{cases}$

$g(B,\{C\}) = d(B,C) + d(C,A) = 4+\infty = \infty$

$g(C,\{B\}) = d(C,B) + d(B,A) = 4+2 = 6$

∴ Minimum cost path is A-B-D-C-E-A / A-E-C-D-B-A

Algorithm:

if $|S| = 2$ then
{
S must be $\{1, i\}$

$C(S, i) = d(1, i)$

}

else {

if $(|S|) > 2$

$C(S, i) = \min\{C(S - \{i\}, j) + d(j, i)\}$

} where j belongs to S, $j != i$ and $j != 1$.

* Using the following formula we can solve the Travelling sales man problem

$$g(i, S) = \min\{d(i, j) + g(j, \{S - j\})\}, \ j \in S$$

where S is the set of nodes of the graph that are not yet traverse and j is a node, $j \in S$.

→ follow the below steps to solve the problem using Dynamic programming.

step 1: Let the function $C(1, V - \{1\})$ is the total length of the four terminating at one. Let $d(i, j)$ be the shortest path b/w two vertices i & j.

step 2: Let v_1 to v_n be the sequence of vertices followed in optimal view then, $v_1, v_2, ---- v_n$ must be the shortest path from v_1 to v_n which passes through each vertex exactly once.

step 3: The following formula can be used to obtain the optimal cost

$$C(i, S) = \min\{d(i, j) + C(j, S - \{j\})\}$$

where $i \notin S$ and $j \in S$

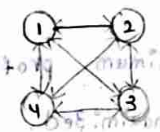
Time complexity:

* Till now, researchers haven't found a polynomial time algorithm for TSP.

→ If we solve TSP using the most direct method (Brute force method) we have to list all the ordered combinations

i.e., we get $(n-1)!$ combinations then the time complexity is $O(n!)$

Ex:



combinations:

1 2 3 4 1

1 3 2 4 1

1 2 3 4 1

1 4 3 2 1

1 4 2 3 1

1 3 4 2 1

* There are 6 combinations.

Here $n = 4$

$(n-1)! = 3! = 6$.

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

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

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

↓ ↓ ↓ ↓ ↓ ↓

$g(4, \phi)$ $g(3, \phi)$ $g(4, \phi)$ $g(2, \phi)$ $g(3, \phi)$ $g(2, \phi)$

→ Total subproblems - 15 (unique - 12)

Here, the total no. of unique sub problems that we get using DP is $(n-1) \cdot 2^{n-2}$, the time taken to solve each problem is 'n'

→ Among the existing algorithms DP solves the problem in $O(n^2 2^n)$ time where 'n' is the no. of nodes

23-09-19

0/1 Knapsack problem:

Here a knapsack or a bag is given, the maximum capacity of a bag, objects will be given each object 'i' is having a profit p_i and the weight w_i

→ The problem here is to insert the objects into the bag without violating the condition

$$\sum_{i=1}^{n} w_i x_i = M.$$

→ The goal here is to get maximum profit ($\sum_{i=1}^{n} p_i x_i$ maximize) to be maximized.

Greedy knapsack (vs) DP knapsack:

* In the greedy method, the objects are completely inserted or a fraction of the object is inserted or the object is not at all inserted. i.e., the values of the objects are 0 (or) 1 (or) a fraction between 0 to 1 (0, 1, 0 ≤ frac ≤ 1)

* In the dynamic programming the object is either completely inserted (or) not at all. But the fraction of the object is not inserted i.e., the values of the objects are either 0 (or) 1.

Formulae:

To solve a 0/1 knapsack using DP the following formula is used.

$f_i(y_j)$ Let $f_i(y_j)$ be the optimal solution.

$$f_i(y) = \max\{f_{i-1}(y),\ f_{i-1}(y-w_i)+p_i\}$$

$$S^0 = \{(0,0)\}$$

$$S_i^0 = \{(P,W) \mid (P-p_i,\ W-w_i) \in S^i\}$$

S^{i+1} can be computed by merging S^i and S_1^i

* Purging rule (or) Dominance rule:

→ If S^{i+1} contains the two pairs (P_j, w_j) and (P_K, W_K) such that $P_j \le P_K$ and $w_j \ge w_K$ then P_j, w_j can be eliminated.

→ It starts removing the pair with less profit and more weight.

24-09-19

Problem: Find the optimal solution.

n	w	p
1	2	1
2	3	2
3	4	5

M = 6.

Ans:

$S^0 = \{(0,0)\}$

$S^{i+1} = S^i + S_1^i$ (Merge)

$S_1^i = S^i + \{P_{i+1}, w_{i+1}\}$ (Addition)

1) i = 0

$S^{i+1} = S^{0+1} = S^1 = S^0 + S_1^0$

$S_1^0 = S^0 + \{(P_1, w_1)\}$

$= (0,0) + (1,2) = (1,2)$

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

→ No pair satisfied the purging rule.

2) i = 1

$S^{i+1} = S^2 = S^1 + S_1^1$

$S_1^1 = S^1 + (P_2, w_2)$

$= \{(0,0)\ (1,2)\} + \{(2,3)\}$

$= \{(2,3)\ (3,5)\}$

$S^2 = \{(0,0)\ (1,2)\ (2,3)\ (3,5)\}$

Here, no pair satisfied the purging rule.

3) $i=2$

$S^{2+1} = S^3 = S^2 + S_1^2$

$S_1^2 = S^2 + (P_3, w_3)$

$= S^2 + (5,4)$

$= \{(5,4)\ (6,6)\ (7,7)\ (8,9)\}$

while merging S_1^2 and S^2 we have the pairs $(3,5)$ and $(5,4)$ satisfies $P_j \le P_k$ and $w_j \ge w_k$

P_j, w_j $\quad$ P_k, w_k

so remove the pair $(3,5)$

$\Rightarrow$ then $S^3 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

To get the optimal solution $w_i x_i \le m$

Here in $(6,6)$ the weight $6 \le 6$ satisfies the condition with maximum profit 6.

To find the objects

$\rightarrow$ Repeat

{

if $(P,w) \in S^{n-1}$

set $x_n = 0$

else

{

set $x_n = 1$

$(P,w) = (P-P_n,\ w-w_n)$

}

$n = n-1;$

} until $(n==0)$

- $n=3$

$(6,6) \notin S^2 \rightarrow$ false

$x_3 = 1.$

$(P,w) = (6-P_3,\ 6-w_3)$

$= (6-5,\ 6-4) = (1,2)$

- $n=2$

$(1,2) \in S^1 \rightarrow$ true

$x_2 = 0$

- $n=1$

$(1,2) \notin S^0 \rightarrow$ false

$x_1 = 1$

$(P,w) = (1-P_1,\ 2-w_2)$

$= (1-1,\ 2-2) = (0,0)$

- $n=0 \rightarrow$ true (exit the loop)

$\therefore$ The objects are $x_1 = 1$, $x_3 = 1$.

2) Find the optimal solution

n	P	w
1	1	2
2	2	3
3	5	4
4	6	5

$M=8$

Ans:

$S^0 = \{(0,0)\}$

$S^{i+1} = S^i + S_1^i$

$S_1^i = S^i + \{(P_{i+1},\ w_{i+1})\}$

1) $i=0$

$S^{i+1} = S^{0+1} = S^1 = S^0 + S_1^0$

$S_1^0 = S^0 + \{(P_1, w_1)\}$

$s_1^0 = s^0 + \{(1,2)\}$

$s_1^0 = \{(1,2)\}$

$s^1 = \{(0,0)\ (1,2)\}$ → No pair satisfied the purging rule.

$i = 1$

$s^2 = s^1 + s_1^1$

$s_1^1 = s^1 + \{(P_2, w_2)\}$

$= \{(0,0)\ (1,2)\} + \{(2,3)\}$

$= \{(2,3)\ (3,5)\}$

$s^2 = \{(0,0)\ (1,2)\ (2,3)\ (3,5)\}$

No pair satisfied the purging rule.

$i = 2$

$s^3 = s^2 + s_1^2$

$s_1^2 = s^2 + \{(P_3, w_3)\}$

$= \{(0,0)\ (1,2)\ (2,3)\ (3,5)\} + \{(5,4)\}$

$s_1^2 = \{(5,4)\ (6,6)\ (7,7)\ (8,9)\}$

$s^3 = \{(0,0)\ (1,2)\ (2,3)\ (3,5)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

Applying purging rule:

$\underset{p_j\ w_j}{(3,5)}$ and $\underset{p_k\ w_k}{(5,4)}$

Here $p_j \le p_k$ and $w_j \ge w_k$ so eliminate $(3,5)$

~~(6,6) and (7,7)~~

~~$p_j\ w_j$~~ ~~$p_k\ w_k$~~

~~Here $p_j \le p_k$~~

$\therefore s^3 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

$s^4 = s^3 + s_1^3$

$s_1^3 = s^3 + \{(P_4, w_4)\}$

$= \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\} + \{(6,5)\}$

$= \{(6,5)\ (7,7)\ (8,8)\ (11,9)\ (12,11)\ (13,12)\ (14,14)\}$

$s^4 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ \cancel{(6,6)}\ \cancel{(7,7)}\ \cancel{(8,9)}\ (6,5)$
$(7,7)\ (8,8)\ (11,9)\ (12,11)\ (13,12)\ (14,14)\}$

<u>purging rule</u>:

$\underset{p_j\ w_j}{(6,6)}$ $\underset{p_k\ w_k}{(6,5)}$

$p_j \le p_k$ and $w_j \ge w_k$ so eliminate $(6,6)$

$\underset{p_j\ w_j}{(7,7)}$ $\underset{p_k\ w_k}{(7,7)}$

$p_j \le p_k$ and $w_j \ge w_k$ so eliminate $(7,7)$

$\underset{p_j\ w_j}{(8,9)}$ and $\underset{p_k\ w_k}{(8,8)}$

$p_j \le p_k$ and $w_j \ge w_k$ so eliminate $(8,9)$

$\therefore s^4 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,5)\ (7,7)\ (8,8)$
$(11,9)\ (12,11)\ (13,12)\ (14,14)\}$

The optimal solution $(8,8)$

Here, the weight is 8 which equal to capacity so it is the max profit 8.

objects to be inserted:

- $n = 4$

$(8,8) \in s^3 \longrightarrow$ false

$x_4 = 1$

$(P, w) = (8 - P_4,\ 8 - w_4)$

$= (8-6,\ 8-5)\ \ (2,3)$

purging rule :

1) $(31,22)$ $(33,15)$
$p_j \; w_j$ $p_k \; w_k$

Here $p_j \le p_k$ and $w_j \ge w_k$ so eliminate the pair $(31,22)$

2) $(52,33)$ $(54,26)$
$p_j \; w_j$ $p_k \; w_k$

Here $p_j \le p_k$ and $w_j \ge w_k$ so eliminate the pair $(52,33)$

3) $(63,35)$ $(65,28)$
$p_j \; w_j$ $p_k \; w_k$

Here $p_j \le p_k$ $w_j \ge w_k$ so eliminate the pair $(63,35)$

$\therefore S^4 = \{(0,0)\ (11,2)\ (21,11)\ (32,13)\ (42,24)\ (33,15)\ (44,27)\ (54,26)\ (65,28)\ (64,37)\ (75,39)\ (85,48)\ (96,50)\}$

Here the optimal solution is $\underline{(75,39)}$

- $n=4$

$(75,39) \in S^3 \rightarrow$ false $\Rightarrow$ $x_4 = 1$

$(p,w) = (75 - p_4,\ 39 - w_4)$
$= (75-33,\ 39-15) = (42,24)$

- $n=3$

$(42,24) \in S^2 \rightarrow$ false $\rightarrow$ $x_3 = 1$

$(p,w) = (42 - p_3,\ 24 - w_3)$
$= (42-31,\ 24-22)$
$= (11,2)$

- $n=2$

$(11,2) \in S^1 \rightarrow$ true

$x_2 = 0$

- ~~$n=1$~~

~~$(0,0) \in S^0 \rightarrow$ true~~

~~$x_1 = 0$~~

- ~~$n=0 \rightarrow$ true (loop exit)~~

~~$\therefore$ The objects are 3 and 4~~

~~$\therefore$ max profit is $\underline{64}$~~

- $n=1$,

$(11,2) \in S^0 \rightarrow$ false

$x_1 = 1$

$(p,w) = (11 - p_1,\ 2 - p_1)$
$= (11-11,\ 2-2)$
$= (0,0)$

- $n=0 \rightarrow$ true (loop exits)

$\therefore$ The profit is 75

and the objects are 1, 3 and 4.

25-09-19

Algorithm for Dynamic knapsack problem :

Algorithm DKP (P, w, n, m)

{

$s^0 = \{(0,0)\}$

for i = 0 to n do

{

$s_1^i = s^i + \{(p_{i+1}, w_{i+1})\}$

s^{i+1} = mergepurge (s^i, s_1^i)

}

let (P,w) is a pair in s^n whose m−w is minimum.

Repeat

{ if $(P,w) \notin s^{n-1}$

$x_n = 0$;

else

{ $x_n = 1$;

$(P,w) = (P-P_n, w-w_n)$;

}

n = n−1 ;

} until (n==0)

}

Time complexity :

26-09-19

Optimal Binary Search Tree :

* A Binary search tree is a Binary tree whose elements in the left subtree are lesser than the root and the elements in the right subtree are greater than the root.
* The elements in the Binary search tree are arranged like this to make searching easier.
* The no.of possible BST's with n elements

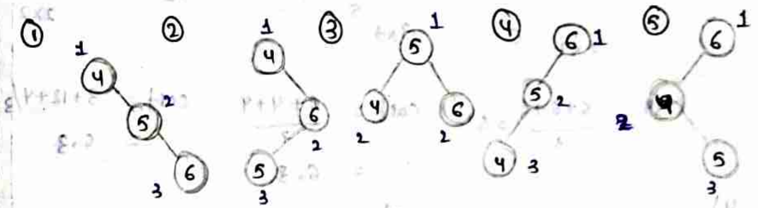
$$\frac{2nC_n}{n+1}$$

Ex: Let us consider the elements 4,5,6 and construct all the possible BST's.

Ans: No.of possible BST's $= \frac{2nC_n}{n+1}$

$$= \frac{2\times3\,C_3}{3+1}$$

$$= \frac{6C_3}{4} = \frac{6\times5\times4}{3\times2\times1} \times \frac{1}{4} = 5$$



* In the 1st BST the No.of comparisons needed to search 4 is 1, to search element 5 is 2 and to search element 6 is 3.

The average comparisons for the 1st BST $= \frac{1+2+3}{3} = 2$

BST 2 : 2 BST 3 : $\frac{5}{3} = 1.6$

BST 4 : 2 BST 5 : 2

27/01/...

Finding the cost of successful and unsuccessful search of a binary tree:

Here the elements will be given, the frequency/probabilities will be given (frequencies of successful search elements) and the frequencies of elements that are not there in the list are also be given (frequencies of unsuccessful search)

we have to find out the cost of OBST.

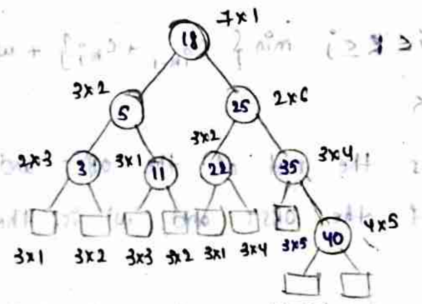
Ex: Elements = (3, 5, 11, 18, 22, 25, 35, 40)

$p(1:8) = (2, 3, 1, 7, 2, 6, 4, 5)$

$q(0:8) = (1, 2, 3, 2, 1, 4, 5, 3, 1)$

Construct the OBST.

Ans:



cost: 7 + 6 + 12 + 6 + 8 + 6 + 12 + 3 + 6 + 9 + 6 + 3 + 12 + 15 + 20 + 12 + 4 = **142**

Ex 2: Construct the OBST

$(a_1, a_2, a_3, a_4) = (\text{do}, \text{if}, \text{int}, \text{while})$

$p(1:4) = (3, 3, 1, 1)$

$q(0:4) = (2, 3, 1, 1, 1)$

Ans: Given that,

0	1	2	3	4
$w_{00} = 2$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = 3$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = 1$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1$ $c_{44} = 0$ $r_{44} = 0$
$w_{01} = 8$ $c_{01} = 8$ $r_{01} = \text{do}$ (1)	$w_{12} = 7$ $c_{12} = 7$ $r_{12} = \text{if}$ (2)	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = \text{int}$ (3)	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = \text{while}$ (4)	
$w_{02} = 12$ $c_{02} = 19$ $r_{02} = \text{do}$ (1)	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = \text{if}$ (2)	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = \text{int}$ (3)		
$w_{03} = 14$ $c_{03} = 25$ $r_{03} = \text{if}$ (2)	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = \text{if}$ (2)			
$w_{04} = 16$ $c_{04} = 32$ $r_{04} = \text{if}$ (2)				

* $w_{ii} = q_i$

$\Rightarrow w_{00} = 2,\ w_{11} = 3,\ w_{22} = 1,\ w_{33} = 1,\ w_{44} = 1$

* $c_{ii} = 0$

$\rightarrow c_{00} = 0,\ c_{11} = 0,\ c_{22} = 0,\ c_{33} = 0,\ c_{44} = 0.$

* $r_{ii} = 0$

$\rightarrow r_{00} = 0,\ r_{11} = 0,\ r_{22} = 0,\ r_{33} = 0,\ r_{44} = 0.$

* $w_{i\,i+1} = q_i + q_{i+1} + p_{i+1}$, $c_{i\,i+1} = q_i + q_{i+1} + p_{i+1}$, $r_{i\,i+1} = i+1$

$w_{01} = q_0 + q_1 + p_1$, $c_{01} = q_0 + q_1 + p_1$

$= 2 + 3 + 3$, $= 2 + 3 + 3 = 8$. $r_{01} = 1$,

$w_{12} = q_1 + q_2 + p_2$ $\qquad r_{12} = 2$

$= 3 + 1 + 3 = 7 \rightarrow c_{12} = 7.$

$w_{23} = q_2 + q_3 + p_3$ $\qquad r_{23} = 3$

$= 1 + 1 + 1 = 3 \Rightarrow c_{23} = 3$

$w_{34} = q_3 + q_4 + p_4$ $\qquad r_{34} = 4$

$\rightarrow c_{34} = 3$

$*\ w_{ij} = w_{ij-1} + p_j + q_j$

$w_{02} = w_{01} + p_2 + q_2$

$= 8+3+1 = 12$

$*\ c_{ij} = \min_{i<k\le j} \{ c_{ik-1} + c_{kj} \} + w_{ij}$

$c_{02} = \min_{0<1\le 2} \{ (c_{00} + c_{12}) \overset{(c_{01}+c_{22})}{} \} + w_{02}$

$= \min \{ (0+7)\ (8+0) \} + 12$

$= 7+12 = 19$ at $k=1$.

$r_{02} = k = 1$.

$*\ w_{ij} = w_{ij-1} + p_j + q_j$;

$w_{13} = w_{12} + p_3 + q_3$

$= 7+1+1 = 9$.

$w_{24} = w_{23} + p_4 + q_4$

$= 3+1+1 = 5$

$w_{03} = w_{02} + p_3 + q_3$

$= 12+1+1 = 14$

$w_{14} = w_{13} + p_4 + q_4$

$= 9+1+1 = 11$

$w_{04} = w_{03} + p_4 + q_4$

$= 14+1+1 = 16$

$*\ c_{ij} = \min_{i<k\le j} \{ c_{ik-1} + c_{kj} \} + w_{ij}$

$c_{13} = \min_{k=2,3} \{ (c_{11} + c_{23}), (c_{12} + c_{33}) \} + w_{ij}$

$= \min \{ (0+3), (7+0) \} + w_{13}$

$= 3+9 = 12$ at $k=2$

$r_{13} = 2$

$c_{24} = \min_{k=3,4} \{ (c_{22} + c_{34}), (c_{23} + c_{44}) \} + w_{ij}$

$= \min \{ (0+3), (3+0) \} + w_{24}$

$= 3+5 = 8$

$r_{24} = 3$

- $c_{03} = \min \{ (c_{00} + c_{13}), (c_{01} + c_{23}), (c_{02} + c_{33}) \} + w_{03}$

$k=1,2,3$

$= \min \{ (0+12), (8+3), (19+0) \} + 14$

$= 11 + 14 = 25$ at $k=2$

$r_{03} = 2$.

- $c_{14} = \min \{ (c_{11} + c_{24}), (c_{12} + c_{34}), (c_{13} + c_{44}) \} + w_{14}$

$k=2,3,4$ $= \min \{ (0+8)\ (7+3)\ (12+0) \} + 11$

$= 8+11 = 19$ at $k=2$

$r_{14} = 2$.

- $c_{04} = \min \{ (c_{00} + c_{14})\ (c_{01} + c_{24})\ (c_{02} + c_{34})\ (c_{03} + c_{44}) \} + w_{04}$

$k=1,2,3,4$ $= \min \{ (0+19)\ (8+8)\ (19+3)\ (25+0) \} + 16$

$= 16+16 = 32$ at $k=2$.

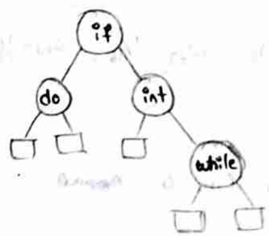
$r_{04} = 2$.

$r_{ij} = k$, $r_{ik-1} \rightarrow$ left $r_{kj} \rightarrow$ right

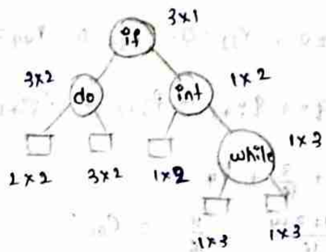
$r_{04} = 2$, $r_{01} \rightarrow 1$ $r_{24} \rightarrow 3$.

$r_{01} = 1$, $r_{00} \rightarrow 0$, $r_{11} = 0$

$r_{24} = 3$ $r_{22} \rightarrow 0$, $r_{34} = 4$.



Cross check :



cost = 3+6+2
+ 4+6+2
+ 3+3+3
= 32

Ex3: Construct

Elements : a_1, a_2, a_3, a_4

$p(1:4) = \left(\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}\right)$

$q(0:4) = \left(\frac{1}{4}, \frac{3}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}\right)$

Ans:

0	1	2	3	4
$w_{00} = 1/4$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = 3/16$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = 1/16$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1/16$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1/16$ $c_{44} = 0$ $r_{44} = 0$
$w_{01} = 11/16$ $c_{01} = 11/16$ $r_{01} = 1$	$w_{12} = 6/16$ $c_{12} = 6/16$ $r_{12} = 2$	$w_{23} = 3/16$ $c_{23} = 3/16$ $r_{23} = 3$	$w_{34} = 3/16$ $c_{34} = 3/16$ $r_{34} = 4$	
$w_{02} = 14/16$ $c_{02} = 20/16$ $r_{02} = 1$	$w_{13} = 8/16$ $c_{13} = 11/16$ $r_{13} = 2$	$w_{24} = 5/16$ $c_{24} = 8/16$ $r_{24} = 3$		
$w_{03} = 16/16$ $c_{03} = 27/16$ $r_{03} = 1$	$w_{14} = 10/16$ $c_{14} = 18/16$ $r_{14} = 2$			
$w_{04} = 18/16$ $c_{04} = 36/16$ $r_{04} = 1$				

→ * $w_{ii} = q_i$

→ $w_{00} = 1/4$, $w_{11} = 3/16$, $w_{22} = 1/16$, $w_{33} = 1/16$, $w_{44} = 1/16$

→ * $c_{ii} = 0$

$c_{00} = 0$, $c_{11} = 0$, $c_{22} = 0$, $c_{33} = 0$, $c_{44} = 0$,

→ * $r_{ii} = 0$

$r_{00} = 0$, $r_{11} = 0$, $r_{22} = 0$, $r_{33} = 0$, $r_{44} = 0$

→ * $w_{i\,i+1} = q_i + q_{i+1} + p_{i+1} = c_{i\,i+1}$

$w_{01} = 1/4 + \frac{3}{16} + 1/4$

$= \frac{4+3+4}{16} = \frac{11}{16} = c_{01}$

$w_{12} = q_1 + q_2 + p_2$

$= \frac{3}{16} + \frac{1}{16} + \frac{1}{8} = \frac{3+1+2}{16} = \frac{6}{16} = c_{12}$

$w_{23} = q_2 + q_3 + p_3$

$= \frac{1}{16} + \frac{1}{16} + \frac{1}{16} = \frac{3}{16} = c_{23}$

$w_{34} = q_3 + q_4 + p_4$

$= \frac{1}{16} + \frac{1}{16} + \frac{1}{16} = \frac{3}{16} = c_{34}$

$r_{i\,i+1} = i+1$

$\Rightarrow r_{01} = 1$, $r_{12} = 2$, $r_{23} = 3$, $r_{34} = 4$

* $w_{ij} = w_{i\,j-1} + p_j + q_j$

$w_{02} = w_{01} + p_2 + q_2$

$= \frac{11}{16} + \frac{1}{8} + \frac{1}{16} = \frac{4+2+1}{16} = \frac{14}{16}$

$w_{13} = w_{12} + p_3 + q_3$

$= \frac{6}{16} + \frac{1}{16} + \frac{1}{16} = \frac{8}{16}$

$w_{24} = w_{23} + p_4 + q_4$

$= 3/16 + 1/16 + 1/16 = 5/16$

$w_{03} = w_{02} + p_3 + q_3$

$= 14/16 + \frac{1}{16} + \frac{1}{16} = \frac{16}{16}$

$w_{14} = w_{13} + p_4 + q_4$

$= \frac{8}{16} + \frac{1}{16} + \frac{1}{16} = \frac{10}{16}$

$w_{04} = w_{03} + p_4 + q_4$

$= \frac{16}{16} + \frac{1}{16} + \frac{1}{16} = \frac{18}{16}$

* $c_{ij} = \min_{i<k\le j} \{ c_{i\,k-1} + c_{kj} \} + w_{ij}$

$c_{02} = \min \{ (c_{00} + c_{12}), (c_{01} + c_{22}) \} + w_{02}$

$C_{02} = \min\{(0+6/16)\ (14/16+0)\}$

$= \frac{6}{16} + \frac{14}{16} = \frac{20}{16}$ • $r_{02} = 1$

$C_{13} = \min_{2,3}\{(C_{11}+C_{23})\ (C_{12}+C_{33})\} + w_{13}$

$= \min\{(0+3/16)\ (6/16+0)\} + 8/16$

$= \frac{3}{16} + \frac{8}{16} = \frac{11}{16}$ • $r_{13} = 2$

$C_{24} = \min_{3,4}\{(C_{22}+C_{34})\ (C_{23}+C_{44})\} + w_{24}$

$= \min\{(0+3/16)\ (3/16+0)\} + 5/16$

$= 3/16 + \frac{5}{16} = \frac{8}{16}$ • $r_{24} = 3$

$C_{03} = \min_{1,2,3}\{(C_{00}+C_{13})\ (C_{01}+C_{23})\ (C_{02}+C_{33})\} + w_{03}$

$= \min\{(0+11/16)\ (11/16+3/16)\ (20/16+0)\} + 16/16$

$= \frac{11}{16} + \frac{16}{16} = \frac{27}{16}$ at $k=1$. • $r_{03} = 1$.

$C_{14} = \min_{2,3,4}\{(C_{11}+C_{24})\ (C_{12}+C_{34})\ (C_{13}+C_{44})\} + w_{14}$

$= \min\{(0+8/16)\ (6/16+3/16)\ (11/16+0)\} + 10/16$

$= \frac{8}{16} + \frac{10}{16} = \frac{18}{16}$ at $k=2$ • $r_{14} = 2$.

$C_{04} = \min_{1,2,3,4}\{(C_{00}+C_{14})\ (C_{01}+C_{24})\ (C_{02}+C_{34})\ (C_{03}+C_{44})\} + w_{04}$

$= \min\{(0+18/16)\ (4/16+8/16)\ (20/16+3/16)\ (27/16+0)\} + 18/16$

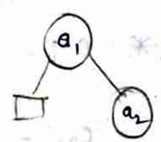
$= \frac{18}{16} + \frac{18}{16} = \frac{36}{16}$ at $k=1$. • $r_{04} = 1$.

• $r_{04} = 1$.

$r_{ij} = k$, $r_{ik-1} \rightarrow$ left, $r_{kj} \rightarrow$ right

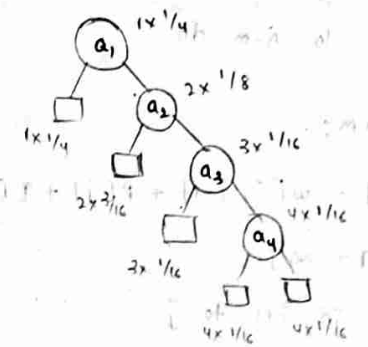
$r_{04} = 1$, $r_{00} \rightarrow 0$, $r_{14} \rightarrow 2$

$r_{14} = 2$, $r_{11} \rightarrow 0$, $r_{24} \rightarrow 3$



$r_{34} = 4$, $r_{33} = 0$, $r_{44} = 0$

∴ Final tree



Cross check :

$\frac{1}{4} + \frac{1}{4} + \frac{2}{8} + \frac{6}{16} + \frac{3}{16} + \frac{3}{16} + \frac{4}{16} + \frac{4}{16} + \frac{4}{16}$

$= \frac{4+4+4+6+3+3+4+4+4}{16} = \frac{36}{16}$

01/10/19 *

Algorithm :

```
Algorithm OBST ( p, q, n )
{  // p is the probabilities of successful search elements
   // q is the frequencies of unsuccessful search elements
   // n is the tree elements
   // rij is the root of the tree/ subtree
   for i:=0 to n-1 do
   {
      w[i,i] = q[i];
      c[i,i] = 0.0;
      r[i,i] = 0;

      w[i,i+1] = q[i]+ q[i+1] + p[i+1];
      r[i,i+1] = i+1;
      c[i,i+1] = q[i] + q[i+1] + p[i+1];
   }
   w[n,n] = q[n];
   c[n,n]=0.0;
```

```
r[n,n] = 0;
for m=2 to n do
{ for i=0 to n-m do
  {
    j = i+m;
    w[i,j] = w[i, j-1] + p[j] + q[j];
    c[i,j] = ∞;
    for k in i+1 to j
    {
      t = c[i, k-1] + c[k,j] + w[i,j];
      if (t < c[i,j])
      {
        c[i,j] = t;
        r[i,j] = k;
      }
    }
  write( c[0,n], w[0,n], r[0,n] );
  }
 }
}
```

<u>Time</u> <u>complexity</u>:

The time complexity of OBST is $\underline{O(n^3)}$

Reliability Design:

Here we have to set up a system with some devices, each device has some cost and reliability. To set up a system / to purchase the devices we have some money with us, it is indicated with 'C'. The reliability of a device is the probability that the device works perfectly.

* Let us set up a system with following devices

D	c	r
D_1	30	0.9
D_2	15	0.8
D_3	20	0.5

C = 100

D	c	r	u
D_1	30	0.9	2
D_2	15	0.8	3
D_3	20	0.5	2

Ans: Here we have to set up a system with maximum reliability with the given cost.

Here we have to find the maximum reliability.

$S^0 = \{(1,0)\}$

→ Initially the reliability is '1', because while setting up the devices, we will multiply the reliabilities and add the costs.

$S^1 = S_1^1 \cup S_2^1$

Here S_1^1 → 1 → Device number, 1 → no. of copies

Here the pair contains (reliability, cost)

$S_1^1 = \{(0.9, 30)\}$

To find S_2^1, we have to find out the reliability if we have two copies of device 1.

$S_2^1 = 1 - (1-r_1)^2$ → (since two devices)

$1 - (1-0.9)^2 \Rightarrow 1 - (0.1)^2 \Rightarrow 1 - 0.01 = 0.99$

$S_2^1 = \{(S_r^0 \cdot 0.99,\ S_c^0 + 60)\} = \{(1 \times 0.99,\ 0+60)\}$

$= \{(0.99, 60)\}$

$S^1 = S_1^1 \cup S_2$

$= \{(0.9, 30)\ (0.99, 60)\}$

→ Here no pairs satisfied purging rule

$S^2 = S_1^2 \cup S_2^2 \cup S_3^2$

$S_1^2 = \{(r_2, c_2)\} = \{(0.8, 15)\}\ \{S_r^1, S_c^1\}$

$= \{(0.72, 45), (0.792, 75)\}$

Here, if we have two copies of device 2 then reliability is

$1-(1-r_2)^2 = (1-(1-0.8)^2) = 1-0.04 = 0.96,$

$2 \times c_2 = 30 \Rightarrow (0.96, 30)$

$S_2^2 = \{(S_r^1 \times 0.96, S_c^1 + 30)\}$

$= \{(0.864, 60)\ \cancel{(0.95, 90)}\}$

this would be eliminated (not feasible).

Because to buy device 3 we need ₹20

S_3^2 means we have to maintain three copies of device 2 then reliability is

$1-(1-r_2)^3 = 1-0.008 = 0.992$

$\text{Cost} = 3 \times C_2 = 45 \Rightarrow (0.992, 45)$

$S_3^2 = \{(S_r^1 \times 0.992, S_c^1 + 45)\}$

$= \{(0.8928, 75), \cancel{(0.982, 105)}\}$

This would be eliminated because the cost is 100 only.

$S^2 = \{(0.72, 45)\ (0.792, 75)\ (0.864, 60)\ (0.8928, 75)\}$

Here the pairs $(0.792, 75)$ $(0.864, 60)$ satisfies

$r_j \quad c_j \quad r_k \quad c_k$

the purging rule. since $0.792 < 0.864$ and $75 > 60$

$\Rightarrow S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$

$S^3 = S_1^3 \cup S_2^3$

$S_1^3 = \{(S_r^2 \times 0.5, S_c^2 + 20)\}$

$= \{(0.36, 65)\ (0.432, 80)\ (0.4464, 95)\}$

S_2^3 means we have to maintain two copies of device 3 then reliability is

$1-(1-r_3)^2 = 1-(1-0.5)^2 = 1-0.25 = 0.75$

$\text{cost}: c_3 \times 2 = 40 \Rightarrow (0.75, 40)$

$S_2^3 = (S_r^2 \times 0.75, S_c^2 + 40)$

$= \{(0.54, 85), (0.648, 100), \cancel{(0.6696, 115)}\}$

cost is above 100.

$S^3 = \{(0.36, 65)\ (0.432, 80)\ (0.4464, 95)\ (0.54, 85)\ (0.648, 100)\}$

Here the pairs ~~(0.4464, 95)~~ (0.4464, 95) x (0.54, 85) satisfies the purging rule. so eliminate (0.4464, 95)

$S^3 = \{(0.36, 65)\ (0.432, 80)\ (0.54, 85)\ (0.648, 100)\}$

The optimal pair is (0.648, 100)

<u>Devices</u>:

D_1	D_2	D_3
1	2	2
30	30	40

2)

D	C	r
D_1	30	0.9
D_2	20	0.8
D_3	15	0.5

and $c = 105$

Find the devices that should be set up to get maximum reliability?

Ans: Given

D	C	r	U → (upper bound)
D_1	30	0.9	2
D_2	20	0.8	3
D_3	15	0.5	3

$S^0 = \{(1,0)\}$

$S^1 = S_1^1 \cup S_2^1$

$S_1^1 = \{(S_r^0 \times 0.9,\ S_c^0 + 30)\}$

$= \{(0.9, 30)\}$

S_2^1 to find this, the reliability with two copies of devices 1

$1 - (1-r_1)^2 = 1 - (1-0.9)^2 = 0.99$

cost: $c_1 \times 2 = 60 \Rightarrow (0.99, 60)$.

$S_2^1 = \{(S_r^0 \times 0.99,\ S_c^0 + 60)\}$

$= \{(0.99, 60)\}$

$S^1 = \{(0.9, 30)\ (0.99, 60)\}$

$S^2 = S_1^2 \cup S_2^2 \cup S_3^2$

$S_1^2 = \{(S_r^1 \times r_2,\ S_c^1 + c_2)\}$

$S_1^2 = \{(0.72, 50)\ (0.792, 80)\}$

S_2^2 to find this, the reliability with two copies of device 2.

$1 - (1-r_2)^2 = 1 - (1-0.8)^2 = 0.96$.

cost: $c_2 \times 2 = 40 \Rightarrow (0.96, 40)$

$S_2^2 = \{(S_r^1 \times 0.96,\ S_c^1 + 40)\}$

$= \{(0.864, 70)\ (\cancel{0.9504, 100})\}$

cost is ~~was above~~ 100, we can't buy device 3.

S_3^2 to find this, the reliability with three copies of device 2.

$1 - (1-r_2)^3 = 1 - 0.008 = 0.992$

cost: $3 \times c_2 = 60 \Rightarrow (0.992, 60)$

$S_3^1 = \{(S_r^1 \times 0.992,\ S_c^1 + 60)\}$

$S_3^2 = \{(0.8928, 90)\ (\cancel{0.98208, 120})\}$

The cost is above 105.

$S^2 = \{(0.72, 50)\ (0.792, 80)\ (0.864, 70)\ (0.8928, 90)\}$

Here the pairs $(0.792, 80)$ & $(0.864, 70)$ satisfies purging rule, $0.792 \le 0.864$ & $80 \ge 70$ so eliminate $(0.792, 80)$

$S^2 = \{(0.72, 50)\ (0.864, 70)\ (0.8928, 90)\}$.

$S^3 = S_1^3 \cup S_2^3 \cup S_3^3$

$S_1^3 = \{(S_r^2 \times 0.5,\ S_c^2 + 15)\}$

$= \{(0.36, 65), (0.432, 85), (0.4464, 105)\}$.

S_2^3 to find this, the reliability with two copies of device 3 is.

$1 - (1-r_3)^2 = 1 - (1-0.5)^2 = 1 - 0.25 = 0.75$.

cost: $c_3 \times 2 = 30 \Rightarrow (0.75, 30)$

$S_2^3 = \{(S_r^2 \times 0.75,\ S_c^2 + 30)\}$

$= \{(0.54, 80)\ (0.648, 100)\ (\cancel{0.6696, 120})\}$

cost is above 105.

S_3^3 to find this, the reliability with three copies of device 3 is,

$1 - (1-r_3)^3 = 1 - (1-0.5)^3 = 1 - 0.125 = 0.875$.

cost: $c_3 \times 3 = 45 \Rightarrow (0.875, 45)$

$S_3^3 = \{(S_r^2 \times 0.875,\ S_c^2 + 45)\}$

$= \{(0.63, 95)\ (\cancel{0.756, 115})\ (\cancel{0.7812, 135})\}$

Cost above 105.

$S^3 = \{(0.36, 65)\ (0.432, 85)\ (0.4464, 105)$
$(0.54, 80)\ (0.648, 100)\ (0.63, 95)\}$

Here the pairs (0.432, 85) & ... and (0.4464, 105) & (0.648, 100) satisfies purging rule.

So eliminate the pairs (0.432, 85) (0.4464, 105)

$S^3 = \{ (0.36, 65)\ (0.54, 80)\ (0.648, 100)\ (0.63, 95) \}$.

Here the optimal pair is "(0.648, 100)"

Devices: D_1 D_2 D_3

1 2 2

30 40 30 = 100

* If the reliability of device (i) is r_i, then the reliability of the system is the product of all the reliabilities

i.e., $\prod_{i=1}^{n} r_i$

* Suppose if we have 'm' devices in a stage then the reliability of the stage is

$\boxed{1-(1-r)^m}$

where, m is the no. of devices in the stage.

probability that the stage(i) will work perfectly

$\phi_i m_i = 1-(1-r_i)^{m_i}$

* In the reliability design, we try to maintain the duplicate copy of the devices with the given cost to maximize the reliability.

i.e., $\prod_{i=1}^{n} \phi_i m_i$ subjected to $\sum_{i=1}^{n} \underset{①}{c_i} \underset{②}{m_i} \le \underset{③}{C}$

Here ① is the cost of device i

② is the no. of copies of the device(i) in stage(i)

③ the total cost of the system.

* Maximum copies of any device that we can keep in any stage is

$$u_i = \left\lfloor \frac{C + c_i - \sum_{j=1}^{n} c_j}{c_i} \right\rfloor$$

Here, $\sum_{j=1}^{n} c_j$ is the cost needed to purchase

- atleast one copy of each device.

i.e., $\sum_{j=1}^{n} c_j = c_1 + c_2 + c_3 + \cdots + c_n$

- Remaining cost after purchasing one copy is

i.e., $C - \sum_{j=1}^{n} c_j$

(Total cost - cost needed to purchase one copy)

procedure:

* This problem is solved similar to the 0/1 knapsack in DP.

* Here first of all we have to find out the values $S^0, S^1, S^2, \cdots S^n$ where n is number of devices.

* Here S^0 is nothing but a set having the reliability and cost of the system with no device.

i.e., $S^0 = \{(1,0)\}$

then S^1 is the union of $S_1^1 \cup S_2^1 \cup \cdots S_u^1$

* S_i^1 is a set having the tuples $\{(R, C)\}$ where r is the reliability of D_1 and c is the cost of D_1.

* If the system is having the one copy of D_1 then

$S^1 = (S_1^1)$

* After finding S^n we pick up the tuple which gives the cost nearer / equal to the given cost.

* The reliability of the entire system is the reliability of the tuple that we picked up from S^n

* The no.of copies of each device is derived from the tuple that we picked up from S^n.

* Each device number can be derived through powers of S.

i.e., S^n, S^{n-1}, S^{n-2} ___ S^1.