$C_{11} = P+S-T+U$

$C_{12} = R+T$

$C_{22} = Q+S$

$C_{22} = P+R-Q+U$

$$T(n) = \begin{cases} b & n \le 2 \\ 7T(n/2)+cn^2 & n>2 \end{cases}$$

$$T(n) = O(n^{2.81})$$

$$\left(\begin{array}{cc|cc} 1 & 2 & 3 & 0 \\ 5 & 6 & 7 & 0 \\ \hline 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

Unit-II

Greedy method
General method

Algorithm Greedy (a, n)   a, n => array with n inputs a[1 : n]

{

Solution : = ϕ; ϕ -> empty set

for i : = 1 to n do

{

x : = select (a);   select is a function (a) procedure

if Feasible (Solution, x) then

Solution : = union (solution, x);

}   Feasible is boolean value

return Solution;   union we should ...

}

// Greedy method is straight forward method to solve any problem.

1 Feasible solution

2 optimum solution

3 objective function

4 constraints => condition

2768	=> optimum solution
5x500	1x2000
1x200	1x500
1x50	1x200
1x10	1x50
1x5	1x10
1x2	1x5
1x1	1x2
	1x1

=> The problem have n inputs and we required as constraints statisfication is called feasible solution

maximum profit } optimum solution
minimum cost

knapsack problem

$$\text{maximize} \sum_{1 \le i \le n} p_i x_i$$

$$\text{subject to} \sum_{1 \le i \le n} w_i x_i \le m$$

and $0 \le x_i \le 1$, $1 \le i \le n$

$n=3$ $m=20$, $(P_1, P_2, P_3) = (25, 24, 15)$

$(w_1, w_2, w_3) = (18, 15, 10)$ $x_1 = 1$, $x_2 = \frac{2}{15}$, $x_3 = 0$

(x_1, x_2, x_3)	capacity $\sum w_i x_i$	Profit $\sum p_i x_i$
$(\frac{1}{2}, \frac{1}{3}, \frac{1}{4})$	16.5	24.25
$(1, \frac{2}{15}, 0)$	20	28.2
$(0, \frac{2}{3}, 1)$	20	31
$(0, 1, \frac{1}{2})$	20	31.5

$x_1 = 0$, $x_2 = \frac{2}{3}$, $x_3 = 1$

To change the order

Algorithm Greedyknapsack (m, n)

(24 15 25)
(15 10 18)

$x_1 = 0 \Rightarrow 1$
$x_2 = 0 \Rightarrow \frac{1}{2}$
$x_3 = 0 \Rightarrow 0$

```
{
  for i := 1 to n do x[i] := 0.0;
  u := m;
  for i := 1 to n do
  {
    if (w[i] > u) then break;
    x[i] := 1.0 ; u := u - w[i];
  }
  if (i ≤ n) then x[i] := u/w[i];
}
```

n objects $1 \le i \le n$ Profit P_1, P_2, P_3

$1 - w_i$ m weight $w_1\ w_2\ w_3 \ldots$ W

x_i $0 \le x_i \le 1$

$P_i x_i$

$x_1 = 1$	25	0	0	0	0
$x_2 = 0$	0		24	0	0
$x_3 = 0$	$\frac{0}{25}$	0	0	1	$\frac{15}{15}$

$2 \times \frac{5}{9}$ $x_1 = \frac{1}{2}$ 12.5

0 $x_2 = \frac{1}{3}$ 8

$\frac{15}{28}$ $x_3 = \frac{1}{2}$ 7.5

$\frac{25}{18}$ $\frac{24}{15}$ $\frac{15}{10}$

1.4 1.6 1.5 $\frac{P}{W}$

$x_1 = 0$ 0

$x_2 = 1$ 24

$x_3 = \frac{1}{2}$ 7.5

1 $n = 7$, $m = 15$ $(P_1, P_2, P_3, P_4, P_5, P_6, P_7) =$

$(10, 5, 15, 7, 6, 18, 3)$

$(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2, 3, 5, 7, 1, 4, 1)$

$n = 7$ $m = 15$

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (10, 5, 15, 7, 6, 18, 3)$

$(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2, 3, 5, 7, 1, 4, 1)$

Weight in ascending order

	$\Sigma p_i x_i$	$\Sigma w_i x_i$	W
$x_1 = 1$	$1 \times 10 = 10$	$1 \times 2 = 2$	$2 = 12 - 2 = 10$ (3)
$x_2 = 1$	$1 \times 5 = 5$	$1 \times 3 = 3$	$3 = 11 - 3 = 8$ (4)
$x_3 = 4/5$	$\frac{4}{5} \times 15 = 12$	$\frac{4}{5} \times 5 = 4$	$5 = 4$
$x_4 = 0$	$0 \times 7 = 0$	$0 \times 7 = 0$	7
$x_5 = 1$	$1 \times 6 = 6$	$1 \times 1 = 1$	$1 = 15 - 1 = 14$
			$4 = 8 - 1 = 4$ (5)
			$1 = 14 - 1 = 13$ (2)

$x_6 = 1$	$1 \times 18 = 18$	$1 \times 4 = 4$
$x_7 = 0$	$0 \times 3 = 0$	$0 \times 1 = 0$
	47	15

profit in decending order

	$\Sigma p_i x_i$	$\Sigma w_i x_i$	p	w	
$x_1 = 1$	$1 \times 10 = 10$	$1 \times 2 = 2$	10	2	$6-2=4$ 3
$x_2 = 0$	$0 \times 5 = 0$	$0 \times 3 = 0$	5	3	
$x_3 = 1$	$1 \times 15 = 15$	$1 \times 5 = 5$	15	5	$11-5=6$ 2
$x_4 = \frac{4}{7}$	$\frac{4}{7} \times 7 = 4$	$\frac{4}{7} \times 7 = 4$	7	7	$\frac{4}{7}$ 4
$x_5 = 0$	$0 \times 6 = 0$	$0 \times 1 = 0$	6	1	
$x_6 = 1$	$1 \times 18 = 18$	$1 \times 4 = 4$	18	4	$15-4=11$ 1
$x_7 = 0$	$0 \times 3 = 0$	$0 \times 1 = 0$	3	1	
	47	15			

maximum profit per unit weight

$P\ (10 \quad 5 \quad 15 \quad 7 \quad 6 \quad 18 \quad 3)$

$W\ (2 \quad 3 \quad 5 \quad 7 \quad 1 \quad 4 \quad 1)$

$$\frac{P_i}{W_i} = \left(\frac{10}{2} \quad \frac{5}{3} \quad \frac{15}{5} \quad \frac{7}{7} \quad \frac{6}{1} \quad \frac{18}{4} \quad \frac{3}{1}\right)$$

$$= (5 \quad 1.6 \quad 3 \quad 1 \quad 6 \quad 4.5 \quad 3)$$

$P\ (6 \quad 10 \quad 18 \quad 15 \quad 3 \quad 5 \quad 7)$

$W\ (1 \quad 2 \quad 4 \quad 5 \quad 1 \quad 3 \quad 7)$

		$\Sigma P_i x_i$	$\Sigma w_i x_i$	
x_1	1	$6 \times 1 = 6$	$1 \times 1 = 1$	
x_2	1	$10 \times 1 = 10$	$2 \times 1 = 2$	$15 - 1 = 14$
x_3	1	$18 \times 1 = 18$	$4 \times 1 = 4$	$14 - 2 = 12$
x_4	1	$15 \times 1 = 15$	$5 \times 1 = 5$	$12 - 4 = 8$
x_5	1	$3 \times 1 = 3$	$1 \times 1 = 1$	$8 - 5 = 3$
x_6	$\frac{2}{3}$	$\frac{2}{3} \times 5 = 3.33$	$3 \times \frac{2}{3} = 2$	$3 - 1 = 2$
x_7	0	$0 \times 7 = 0$	$0 \times 7 = 0$	$2 - 2 = 0$
		55.33	15	

The three feasible soultion

$(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$	$\Sigma w_i x_i$	$\Sigma P_i x_i$
$(1, 1, \frac{4}{5}, 0, 1, 1, 1)$	15	54
$(1, 0, 1, 4/7, 0, 1, 0)$	15	47
$(1, 1, 1, 1, 1, \frac{2}{3}, 0)$	15	55.33

min-cost Spanish Tree

Prim's Algorithm

E ⇒ Set of edges
C ⇒ 2 dim adjacency matrix
n ⇒ node
t ⇒ Spanning tree

Algorithm Prim (E, Cost, n, t)

{

Let (k, l) be an edge of minimum cost in E;

min cost := Cost [k, l];

t[1,1] := k; t [1, 2] := l;

for i := 1 to n do

if (cost [i, l] < cost [i, k]) then near [i] := l

else near [i] := k;

near is one dimensional array

near [k] := near [l] := 0;

for i := 2 to n-1 do

{

let j be an index such that

near [j] ≠ 0 and cost

[j, near [j]] is minimum ;

t [i, 1] := j ; t [i, 2] := near [j];

mincost := mincost + cost [j, near [j, near[j]];

near [j] := 0;

for k := 1 to n do

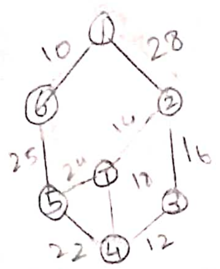
for ((near [k] ≠ 0 and (cost [k, near [k]] >

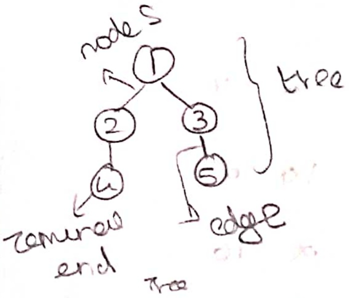
cost [k, j])

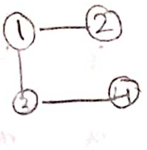
}

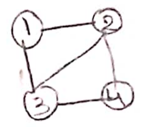
refer min cost

}









Tree

Ordered, loop.

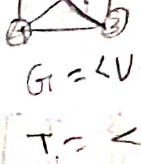
For any other node they only 1 path

n node : n-1 edges ⇒ Tree

All trees called as graph

All graph <u>not</u> called as tree

Both are non-linear are <u>Tree</u> and <u>graph</u>

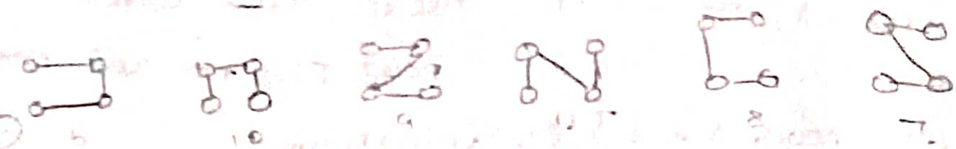


G = <V, E>

T = <V, E'>

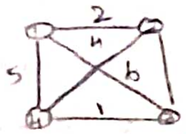
subset

$T \subseteq G$ $\quad E \subseteq E'$

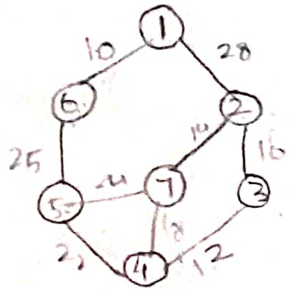


Spanic tree of the graph

. All n vertise have means n-1 a spanic tree



MST - minimum spanic tree



229

$t[1:n-1, 1:2]$

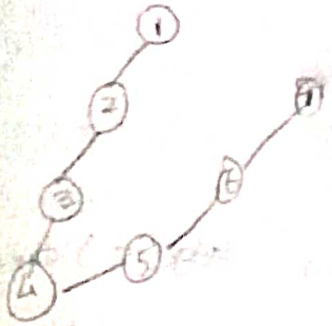
$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & \alpha & 28 & \alpha & \alpha & \alpha & 10 & \alpha \\ 2 & 28 & \alpha & 16 & \alpha & \alpha & \alpha & 10 \\ 3 & \alpha & 16 & \cancel{16} & 18 & \alpha & \alpha & \alpha \\ 4 & \alpha & \alpha & 12 & \alpha & 22 & \alpha & 18 \\ 5 & \alpha & \alpha & \alpha & 22 & \alpha & 25 & 24 \\ 6 & 10 & \alpha & \alpha & \alpha & 25 & \alpha & \alpha \\ 7 & \alpha & 14 & \alpha & 18 & 24 & \alpha & \alpha \end{array}$$

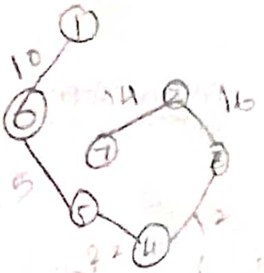
$k=1$

$u=6$

1 $\begin{bmatrix} 2 & 6 \\ 28 & 10 \end{bmatrix}$ 5 $\begin{bmatrix} 4 & 6 & 7 \\ 22 & 25 & 24 \end{bmatrix}$ 3 $\begin{bmatrix} 2 & 4 \\ 16 & 12 \end{bmatrix}$

6 $\begin{bmatrix} 1 & 5 \\ 10 & 25 \end{bmatrix}$ 4 $\begin{bmatrix} 3 & 5 & 7 \\ 12 & 22 & 18 \end{bmatrix}$ 2 $\begin{bmatrix} 1 & 3 & 7 \\ 28 & 16 & 14 \end{bmatrix}$





$t[1,1]=1$ $t[1,2]=6$

$t[2,1]=6$ $t[2,2]=5$

$t[3,1]=5$ $t[3,2]=4$

$t[4,1]=4$ $t[4,2]=3$

$t[5,1]=3$ $t[5,2]=2$

$t[6,1]=2$ $t[6,2]=7$

$10+25$

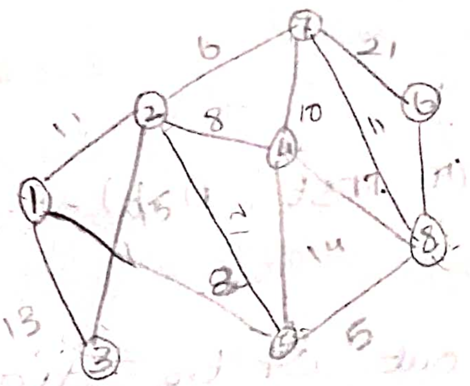
$35+22$

$=57+12$

$=69+16$

$=85+14$

$=99$



13.

Kruskal's Algorithm

t : = 0

while ((t has less than n-1 edges) and
(E ≠ ϕ)) do

{

Choose an edge (v, w) from E of lowest cost.

Delete (v, w) from E;

if (v, w) does not create a cycle in t

than add (v, w) to t ;

else discard (v, w) ;

}

E => set of edges

t => spanning tree

Algorithm Kruskal (E, cost, n, t)

{

construct a heap out of the edge costs
using Heapify.

for i = 1 to n do Parent [i] : = -1 ;

// Each vertex is in a different set

i = 0 ; mincost : = 0.0 ,

while (i < n-1) and (heap not empty)) do

{

Delete a minimum cost edge (u, v) from the
heap and re. using Adjust ;

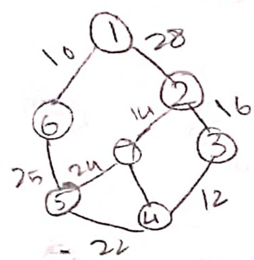
j : = Find (u) ; k : = Find (v) ; Parent

if (j ≠ k) then

{

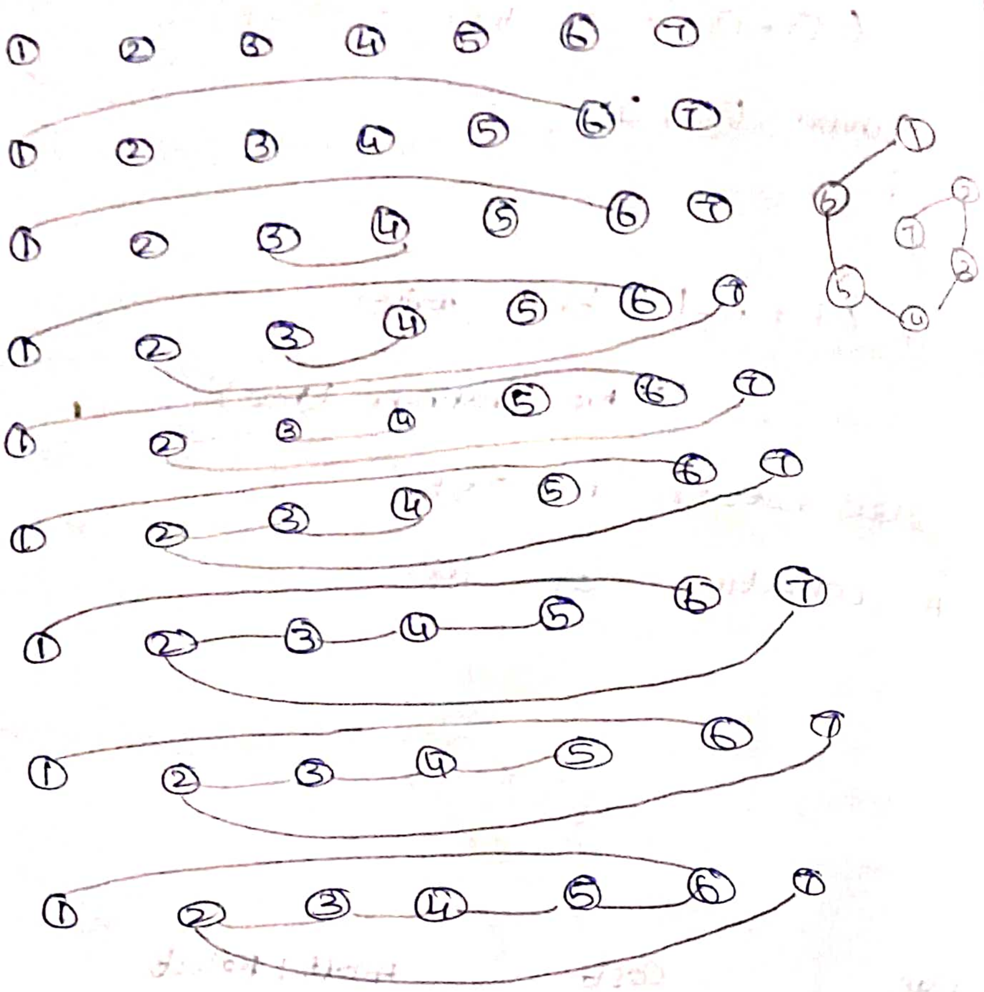
```
i := i+1 ;
t[i,1] := u, t[i,2] := v;
union (j,k);
}
}
if (i ≠ n-1) then write
        ("No spanning tree");
else return mincost;
}
```

A collection of di



Edge	Cost	Accept / Reject
(1,6)	10	Accept
(3,4)	12	Accept
(2,7)	14	Accept
(2,3)	16	Accept
(4,7)	$\boxed{18}$	Reject
(4,5)	22	Accept
(5,7)	$\boxed{24}$	Reject
(5,6)	25	Accept
	99	

Spanning Forest



Job Sequencing with Deadlines

Algorithm Greedy_Job (d, J, n)

// J is a set of jobs that can be

completed by their deadlines

{

J := {1};

for i := 2 to n do

{

if (all jobs in J ∪ {i} can be

completed by their deadlines

then J := J ∪ {i};

}

}

Job is work that can be divide into work

$1 \leq i \leq n$

$d_i \geq 0$ $d_i \Rightarrow$ deadline

$P_i \geq 0$ $P_i \Rightarrow$ Profit

only one unit of time we can use the process Job

$n = 4$

$(P_1, P_2, P_3, P_4) = (100, 10, 15, 27)$

$(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

Feasible sol	Processing Sequence	value
(1,3)	(1,3)	115
(2,1)	(2,1)	110
(2,3)	(2,3)	25
(3,1)	(3,1)	115
(4,1)	(4,1)	127
(4,3)	(4,3)	42
(2)	(2)	10
(1)	(1)	100
(3)	(3)	15
(4)	(4)	27

Profit

1 $n=5$

$(P_1, P_2, P_3, P_4, P_5) = (20, 15, 10, 5, 1)$

$(d_1, d_2, d_3, d_4, d_5) = (2, 2, 1, 3, 3)$

2 $n=7$

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (3, 5, 20, 18, 1, 6, 30)$

$(d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (1, 3, 4, 3, 2, 1, 2)$

1 $n=5$

$(P_1, P_2, P_3, P_4, P_5) = (20, 15, 10, 5, 1)$

$(d_1, d_2, d_3, d_4, d_5) = (2, 2, 1, 3, 3)$

feasible Sol	Processing sequence	value
(1)	(1)	20=20
(1,2)	(1,2)	20+15=35
(1,2)	(1,2)	20+15=35
(1,2,4)	(1,2,4)	20+15+5=40

Optimal Solution of Job Sequence = {1, 2, 4}

Profit = 40

2 $n=7$.

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (3, 5, 20, 18, 1, 6, 30)$

$(d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (1, 3, 4, 3, 2, 1, 2)$

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (30, 20, 18, 6, 5, 3, 1)$

$(d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (2, 4, 3, 1, 3, 1, 2)$

feasible Sol

feasible Sol	Processing Sequence	Value
(1)	(1)	30
(1,2)	(1,2)	30+20=50
(1,3,2)	(1,3,2)	30+18+20=68
(4,1,3,2)	(4,1,3,2)	6+30+18+20= 74

Optimal Solution of Job sequence = (4,1,3,2)

Profit = 74

Dijsktra's

Single Sorce Shortest Path

Algorithm ShortestPaths (v, cost, dist, n) v = source vertex

cost = adjust matrix

Algorithm ShortestPaths (v, cost, dist, n) dist = distance

one dimension

n = no of vertex

{

for i := 1 to n do

{

S[i] = false ; dist [i] = cost [v, i]

}

S [v] = true . dist [v] = 0.0

for num = 2 to n-1, do

{

choose u from among those vertices

not in S such that

dist[u] in minimum

S[u] = true

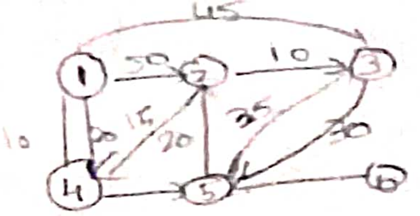
For (each w adjacent to u with S[w]=False

if (dist[w] > dist [u] + cost [u,w]) Then

dist (w) = dist [u] + cost [u, w];

}

}



$V=1$

$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 50 & 45 & 10 & \alpha & \alpha \\ 2 & \alpha & 0 & 10 & 15 & \alpha & \alpha \\ 3 & & & & & & \\ 4 & & & & & & \\ 5 & & & & & & \\ 6 & & & & & & \end{array}$$

$$\begin{array}{lcccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \text{DIST} & (0 & \cancel{50}\ 45 & 45 & 10 & \cancel{\alpha}\ 25 & \alpha) \end{array}$$

$$\begin{array}{lcccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ S & (0 & \cancel{0} & 0 & \cancel{0} & \cancel{0} & 0) \\ & 1 & 1 & & 1 & 1 & \end{array}$$

$u=4$ $\qquad \alpha > 10+15$

$w=1,5$

$u=5$ $\qquad 50>25+20$

$w=2,3$

$u=2$ $\qquad 45>25+35$

$w=3,4$

$u=3$ $\qquad 45>45+10$

$w=5$

optimal storage on tapes

Algorithm Store (n, m)

```
{
  j:=0;
  for i:=1 to n do
  {
    write ("append Program", i, " to Permutation
                               for tape", j);
    j:=(j+1) m = dm;
  }
}
```

n Programs - l_i

$1 \le i \le n \to l_i$

$n = 3$ $(l_1, l_2, l_3) = (5, 10, 3)$

1, 2, 3

ordering (I) length (d I)

1, 2, 3	5+15+18 = 38
1 3 2	5+8+18 = 31
2 1 3	10+15+18 = 43
2 3 1	10+13+18 = 41
3 1 2	3+8+18 = 29
3 2 1	3+13+18 = 34

$l_1 \le l_2 \le l_3 \ldots$

multiple

6, 10, 4, 5, 7, 25, 30, 15, 20, 8

4, 5, 6, 7, 8, 10, 15, 20, 25, 30

1. 4+11+26+56 = 97
2. 5+13+33 = 51
3. 6+16+41 = 63

$\frac{211}{3} = 70.33$