What is a Heap?

What is heap Property?
Heap is Top heavy heap -- MaxHeap A heap is top heavy if the value at any node exceeds the value at its child nodes.
Heap is Bottom heavy heap -- MinHeap A heap is bottom heavy if the value at any node is less than the value at its child nodes.

Is the sequence (23,17,14,6,13,10,1,5,7,12) a max heap? No

Heap – A Heap can be easily implemented as an array A[1…n] such that
A[1] is the **root**,
children of A[i] are ---------
parent of A[i] is ---------

A complete binary tree with n nodes, what is the height h in terms of n

If n is the number of nodes in a heap of height h
What is the Min number of **leaf** nodes?
What is the Max number of **leaf** nodes ?

Show that with the array representation for storing n-element heap, the leaves are the nodes indexed by $\lfloor n/2 \rfloor+1$, $\lfloor n/2 \rfloor+2$, $\lfloor n/2 \rfloor+3$, …, n.

Give an algorithm to create a heap with an array of n elements.
Describe the invariant? Show that invariant holds good at every step
Prove the complexity of creating heap with n nodes is O(n)

Give a criteria to adjust a heap rooted at element i.

AdjustMaxHeap(A,3) on the array A =(27,1,17,3,16,13,10,1,5,7,12,4,8,9,0)

Build a maxHeap on the array A=(5,3,17,10,84,19,6,22,9)

Give an algorithm to heap sort an array with n elements. What is the complexity of heap sort?
Describe the invariant? Show that invariant hold good at every step

What is dynamic programming? Give examples?
What is Greedy Algorithm? Give examples?

What is principle of optimality?

Give recursive algorithm for creating a dynamic table for n weights w and values v, sack size W.
Describe the invariant? Show that invariant hold good at every step

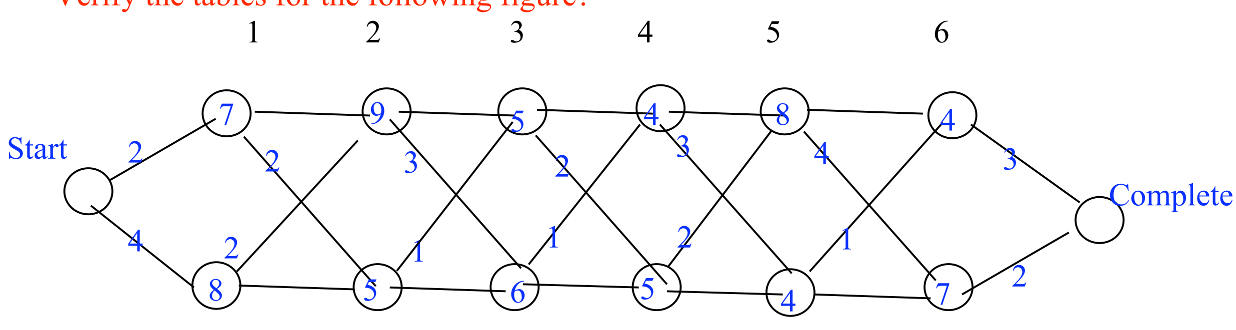
How do you trace the weights used in the Optimal value table.
Describe the invariant? Show that invariant hold good at every step

Example Create table for optimal sack with weights and values.
Example: W =5,
$w_1 = 1$, $w_2 = 2$, $w_3 = 3$
$v_1 = 6$, $v_2 = 10$, $v_3 = 12$

Determine the optimal value and optimal weights.

Find the Longest Common Subsequence in
Give an algorithm to find LCS in
$X_n = (x_1, x_2, \ldots, x_n)$
$Y_m = (y_1, y_2, \ldots, y_m)$
is
$Z_k = (z_1, z_2, \ldots, z_k)$
Find all longest common subsequences (LCS) of
BDCABAO AND ABCBDBA
GRAMPRO and PROGRAM:
10010101 and 010110110
1232412 and 243121
Give an algorithm to determine the length of LCSs.
Give an algorithm to determine the LCSs.

In the Car assembly problem with two assembly lines,
Verify the tables for the following figure?



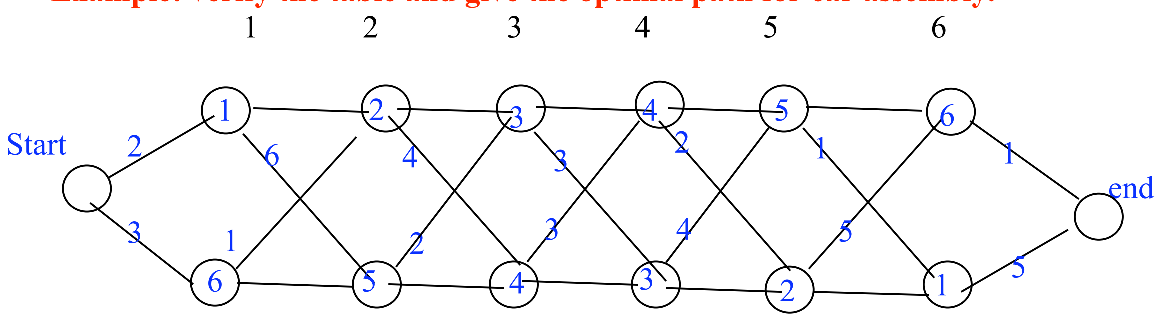
j	1	2	3	4	5	6	exit
$f_1[j]$	9	18	22	26	34	35	5
$f_2[j]$	12	16	22	27	31	38	1

f* = 38, l*=2 – last station is $S_{2,6}$

j	2	3	4	5	6
$l_1[j]$	1	2	1	1	2
$l_2[j]$	1	2	2	2	2

Give the optimal path for car assembly.

Example: verify the table and give the optimal path for car assembly.

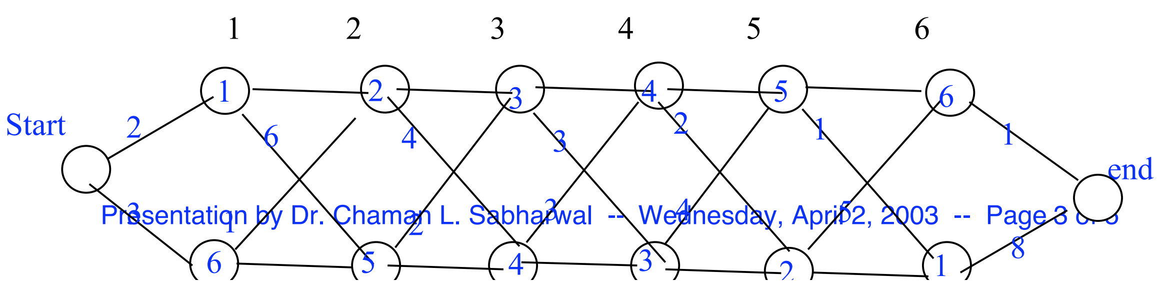


j	1	2	3	4	5	6	exit
$f_1[j]$	3	5	8	12	17	23	1
$f_2[j]$	9	14	13	14	16	17	5

f* = 22, l*=2 – last station is $S_{2,6}$
The sequence of stations is $S_{1,1}$, $S_{1,2}$, $S_{1,3}$, $S_{2,4}$, $S_{2,5}$, $S_{2,6}$

j	2	3	4	5	6
$l_1[j]$	1	1	1	1	1
$l_2[j]$	1,2	1	1	1,2	2

Example same as above with slight change at the end



j	1	2	3	4	5	6	exit
$f_1[j]$	3	5	8	12	17	23	1
$f_2[j]$	9	14	13	14	16	17	8

f* = 24, l*=1 – last station is $S_{1,6}$
The sequence of stations is $S_{1,1}$, $S_{1,2}$, $S_{1,3}$, $S_{1,4}$, $S_{1,5}$, $S_{1,6}$

j	2	3	4	5	6
$l_1[j]$	1	1	1	1	1
$l_2[j]$	1,2	1	1	1,2	2

Is the optimal value unique?
Is the optimal solution unique?
Justify your answers with examples?

- Define

- **Degree** d(v) of a vertex v
  Indegree = # of edges ending at v
  Outdegree = # of edges originating at v

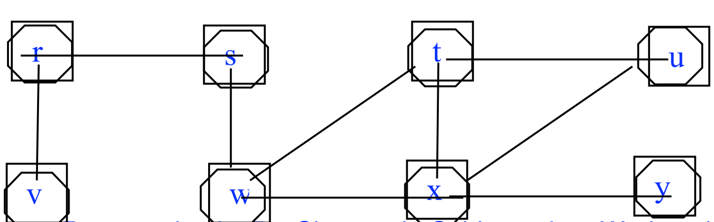
  If G is directed graph, What is the sum of lengths of adjacency lists ?
  If G is undirected graph, what is the sum of lengths of adjacency lists ?

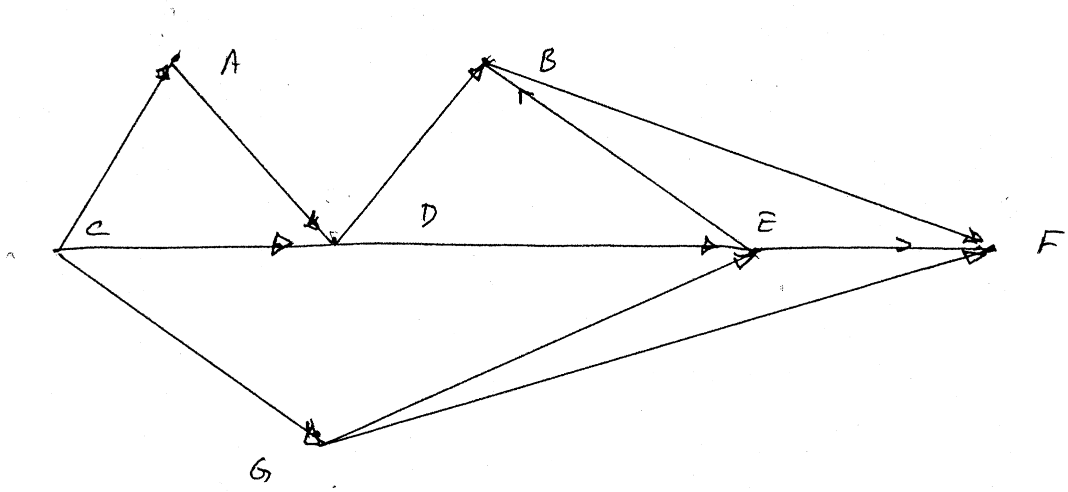
Exercise 22.1-1

a. outdegree of a vertex =O(length of list)
   In matrix representation, row (to the right of i) sum is the outdegree(v_i)
b. indegree of a vertex = O(sum of lengths of lists: |E|) because every list is searched for it.
   In matrix representation, column (below i) sum is the indegree(v_i)

Give advantages of adjacency list over the matrix representation of edge info?
Give advantages of matrix representation over the adjacency list of edge info?

Create BFS numbers for the graph starting at s





Describe an algorithm for topological search. Describe the invariant? Show that invariant hold good at every step