Name: Section: Student No:

Show your work! Give the best result that you can give!
Each question worth 15 points unless stated otherwise

1. Given General Knapsack Problem:

$$\begin{array}{lll} & \max & 7x_1 + x_2 + 4x_3 + 10x_4 + 17x_5 \\ \textbf{MP(b)} & \text{such that} & 3x_1 + x_2 + 2x_3 + 4x_4 + 6x_5 \quad = b \\ & & x_j \in \{0, 1, 2, 3, \ldots\} \quad \forall j \end{array}$$

A solution is given with the Table:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$F(t)$	0	1	4	7	10	11	17	18	21	24	27	28	34	35	38	41	44	45	51	52	55
$p(t)$	–	2	3	1	4	1	5	1	3	1	4	1	5	2	3	5	4	1	5	2	3

Find/Draw the longest path tree!

And write solution for t=17, and t=15 as $X = (x_1, x_2, x_3, x_4, x_5)$.

Solution:

Reformulate the problem as: $F(b) = \max wx : ax = b, x \in N$, where N is the set of natural numbers $\{$0, 1, 2, 3, ... $\}$.

Main observation for the Graph of longest path problem:
for $1 \leq k \leq n$, the edge between (t, t+a_k) with weight w_k.

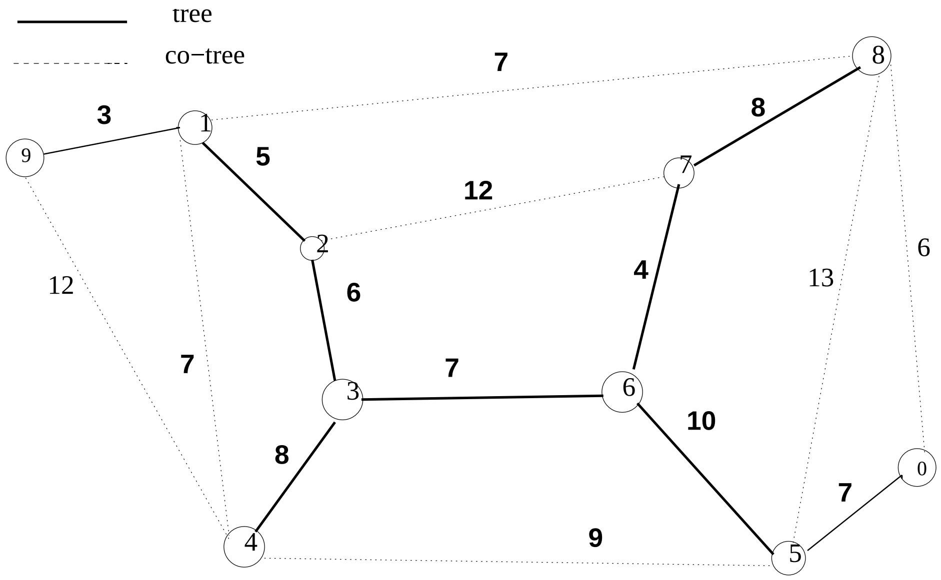
Thus p(t)=k means in the (optimal) Longest Path Tree we have an edge between ($t - a_k$, t) i.e. patent node of t is $t - a_k$. Thus, these edges form the Longest Path Tree.

You can find the r-t path using the above graph, or using the following algorithm:
let X=(0,0,0,0,0)
while $t > 0$
let k=p(t)
$x_k \leftarrow x_k + 1$
$t \leftarrow t - a_k$
endwhile

for t=17, p(17)=1, X=(1,0,0,0,0), and t=t-a_1=17-3=14
t=14, P(14)=3, X=(1,0,1,0,0), and t=t-a_3= 14 -2 =12
t=12, p(12)=5, X=(1,0,1,0,1), and t=t-a_5= 12 - 6 =6
t=6, p(6)=5, X=(1,0,1,0,2) and t=t-6=6-6=0 DONE
Note that ax=3+2+2*6=15 and wx=7+4+2*17=45=F(17)

for t=15, p(15)=5, X=(0,0,0,0,1) and t=t-a_5=15-6=9
t=9, p(9)=1, X=(1,0,0,0,1) and t=t-a_1 = 9 -3 =6
t=6, p(6)=5, X=(1,0,0,0,2) and t=t-6=6-6=0 DONE
Note that ax=3+2*6=15 abd wx=7+2*17=41=F(15)

2. Given the graph with dark edges as tree and dotted edges as co-tree edges:



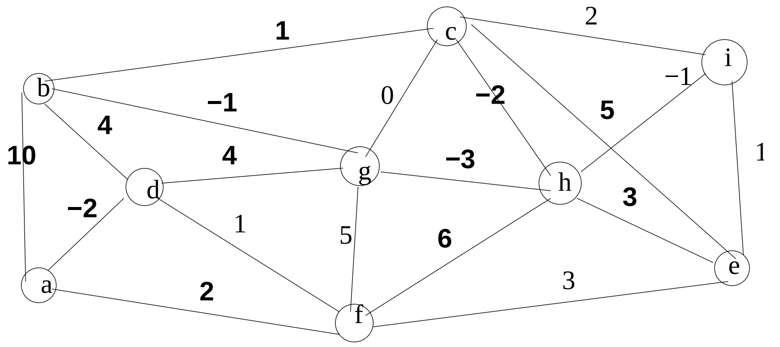
- Determine fundamental Cycle C(T,e) for e=(0,8) (i.e. list the edges) **2 pts**
  C(T,e)={(0,8), (8,7), (7,6), (6,5), (5,0) }
  Note that C(T,e) = path from two ends of edge in T + e
- Determine fundamental Cocyle D(T,f) for f=(2,3) (i.e. list the edges) **2pts**
  D(T,f) = { (1,8), (2,7), (2,3), (1,4), (9,4) }
  T-f has two components: X={ 9, 1, 2 } and X^c = { 3, 4, 5, 6, 7,8, 0 }. D(T,f) is the set of edges between X and X^c
- By using fundamental cocycles, check for optimality of T; and find an optimal tree by switch a tree edge with a co tree edge until finding an optimal tree. Show your calculations **9 pts**

Solution: Optimality of a tree (for minimum spanning tree) in terms of fundamental cycles is: for $e \in T^{\perp}$ and $f \in C(T, e)$ we must have $w_f \leq w_e$.

n terms of cocyles: for $f \in T$, for $e \in D(T, f)$ we must have $w_f \leq w_e$.

Thus we need to check the above inequality for each $f \in T$. In the case $w_f > w_e$ for some $e \in D(T, f)$, we switch switch e and f, i.e T' = T +e - f . We must do this untill no more f remains. It make sense to find e with smallest weight among D(T,f) and switch e with f. Cost of initial tree w(T)=58. We check all $f \in T$ and do the following switches:
f=(4,3), e=(1,4), $w_f = 8 > 7 = w_e$, switch e with f
f=(6,5), e=(0,8), $w_f = 10 > 6 = w_e$, switch e with f
f=(7,8), e=(1,8), $w_f = 8 > 7 = w_e$, switch e with f
Reduction in cost= (8-7)+(10-6)+(8-7)=6. Thus new cost is w(T)=52.

3. Given the graph G with root r, apply Prim-Dijkstra algorithm for the Minimum Spanning Tree Problem (MST). You can implement as you desire. Fill the following table. Where iter means iteration, 'node' means node added to tree, 'edge' means edge=(i,j) with i in X and j in X^c, 'weight' means w(i,j) cost of the edge selected, and 'total' means total cost of the selected edges. Draw the final tree! **Let r=h**



iter.	node	edge	weight	total	parent
0	r	-	-	0	
1					
2					
3					
4					
5					
6					
7					
8					
9					

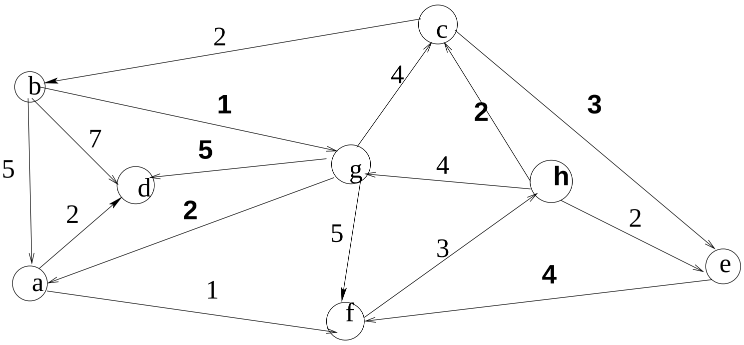
Solution:

We orient tree edges away from r, and this will define parent pointers. Recall that we start T=r and always choose an edge on the boundary of T with minimum weight, and that edge to T and continue.

iter.	node	edge	weight	total	parent
0	r	-	-	0	-
1	g	hg	-3	-3	h
2	c	hc	-2	-5	h
3	i	ih	-1	-6	h
4	b	bg	-1	-7	g
5	e	ie	1	-6	i
6	f	ef	3	-3	e
7	d	fd	1	-2	f
8	a	ad	-2	-4	d
9					

Parent pointers define Min,imum Spanning Tree!

4. Consider the Shortest Path Problem for the directed graph G given below, with selected r=b. Apply 4 iterations of Dijkstra's Shortest Path Algorithm. Fill the following table,



iter	node	edge	parent	dist	total	a	b	c	d	e	f	g	h
						∞	0	∞	∞	∞	∞	∞	∞
0	$r=b$	$-$	$-$	0	0								
1													
2													
3													
4													
5													

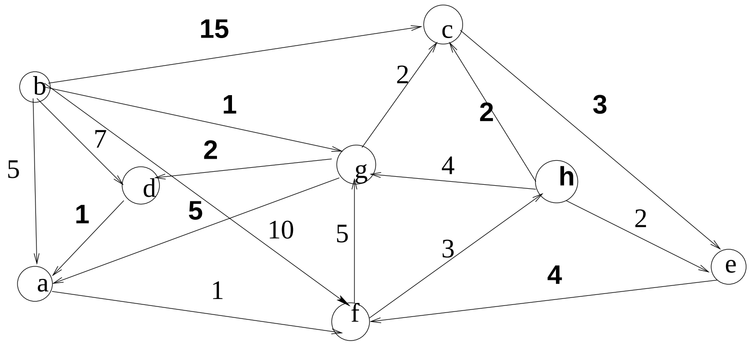
Where 'iter' mean iteration count, 'node' i is the node added and to be scanned, 'edge' is the edge added to Tree, 'parent' is p(i); parent of the node which is added to tree, 'dist' is the distance from r to i (shortest distance), and 'total' is the sum of distances of nodes in the current tree T. The right part of table should contain d(j)/p(j) values. When they are fixed they will move to left part.

Solution:

iter	node	edge	parent	dist	total	a	b	c	d	e	f	g	h
						∞	0	∞	∞	∞	∞	∞	∞
0	$r=b$	$-$	$-$	0	0	$5/b$			$7/b$			**1/b**	
1	g	bg	b	1	1	**3/g**		**5/g**	$6/g$		$6/g$		
2	a	ga	g	3	4				**5/a**		**4/a**		
3	f	f	a	4	8								$7/f$
4	c	cg	g	5	13					$8/c$			
5	d	ad	a	5	18								

Parent pointers defines the Shortest Path Tree.

5. Consider the Shortest Path Problem for the directed graph G given below, with selected r=b. Apply 4 iterations of Bellman-Ford Algorithm. Fill the following table,



iter k	1/a	2/b	3/c	4/d	5/e	6/f	7/g	8/h
$k=0$	∞	0	∞	∞	∞	∞	∞	∞
$k=1$								
$k=2$								
$k=3$								
$k=4$								

Solution:

we use the algorithm: given the vector d^k to obtain d^{k+1}: for each node j
let temp=$\min_i\{d^k(i)+w(i,j)\} = d^k(t)+w(t,j)$
if temp $< d^k(j)$ then $d^{k+1}(j) =$ temp , p(j)=t

iter k	1/a	2/b	3/c	4/d	5/e	6/f	7/g	8/h
$k=0$	∞	0	∞	∞	∞	∞	∞	∞
$k=1$	$5/b$		$15/b$	$7/b$		$10/b$	$1/b$	
$k=2$			$3/g$	$3/g$	$18/c$	$6/a$		$13/f$
$k=3$	$4/d$				$6/c$			$9/f$
$k=4$					$5/a$			
d	4	0	3	3	6	5	1	9
p	d	$-$	g	g	c	a	b	f

6. **Union-Find** Starting with forest of $\{1, 2, \ldots, 15\}$ each with rank=0, apply following union operations a=(5,6), b=(7,8), c=(1,2), d=(3,4), e=(6,8), f=(2,3), g=(9,11), h=(10,12), i=(11,10), j=(10,13), k=(9,2), l=(5,1),m=(8,13) n=(15,5). When taking union of two equal rank sets choose the one whose root has smallest index as the new root. Use also path compression at each iteration. Mark any edge which is cut from the tree with x. **10**

Also write down final rank of the nodes

node	1	3	5	7	9	11	13
rank	3	1	2	1	2	0	0

node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
rank	3	0	1	0	2	0	1	0	2	1	0	0	0	0	0

Let p(i) parent of i, r(i) rank of i, t(i) root or name of the set whih contain i.
let p(i)=i, t(i)=i, r(i)=0 for i=1,...,15

a=(5,6) ⇒p(6)=5, r(5)=1, t(6)=5
b=(7,8) ⇒p(8)=7, r(7)=1 t(8)=7
c=(1,2) ⇒p(2)=1, r(1)=1 t(2)=1
d=(3,4) ⇒p(4)=3, r(3)=1 t(4)=3
e=(6,8) : t(6)= 5, t(8)=7, r(5)=r(7)=1 ⇒ r(5)=2, p(7)=5, t(7)=5
f=(2,3) : t(2)=1, t(3)=3, r(1)=r(3)=1 ⇒ r(1)=2, p(3)=1, t(3)=1
g=(9,11) ⇒ r(9)=1, p(11)=9, t(11)=9
h=(10,12) ⇒ r(10)=1, p(12)=10, t(12)=10
i=(11,10) : t(11)=9 r(9)=r(10)=1 ⇒ r(9)=2 p(10)=9, t(10)=9
j=(10,13) : t(10)=9, r(9)=2, r(13)=0 ⇒ p(13)=9
k=(9,2) : r(9)=2, t(2)=1, r(1)=2 ⇒ r(1)=3, p(9)=1, t(9)=1
l=(5,1) : r(5)=2, r(1)=3 ⇒ p(5)=1, t(5)=1
m=(8,13) : t(8)=1, t(13)=1 ⇒ **Path Compression** find(8): p(8)=p(7)=1; find(13): p(13)=1
n=(15,5) : r(15)=0, t(15)=p(15)=15 ⇒ p(15)=1, t(15)=1
Notice that p(14)=0, r(14)=t(14)=14.

7. **0-1 Knapsack Problem**

Consider the 0-1 Knapsack Problem

$$\mathbf{F(t)} \quad \begin{array}{lll} \max & \sum_{j=1}^{j=n} w_j x_j & \\ \text{such that} & \sum_{j=1}^{j=n} a_j x_j & \leq \quad t \\ & x_j \in \{0,1\}, \text{ for each } j & \end{array}$$

Define $F_k(t)$ as

$$F_k(t) = \max\{\sum_{j=1}^{j=k} w_j x_j : \sum_{j=1}^{j=k} a_j x_j \leq t, x_j \in \{0,1\}\}$$

Assume that $F_k(0) = 0$ for all k, $F_o(t) = 0,$ for $t \geq 0$, and $F_k(t) = -\infty$ for $t < 0, k \geq 0$.

i) Write down a recurrence relation for $F_k(t)$ involving $F_k(.), a_k, w_k, t$ **1 pt**

$$F_k(t) = \max\{F_{k-1}(t), F_{k-1}(t - a_k) + w_k$$

i) Given the following 0-1 Knapsack Problem solve it by applying a recurrence relation and filling the following table. You need to store values of $F_k(t)$, for missing values of k and t. Also you need to store value of x_k in evaluating $F_k(t)$. Show it as $F_k(t)/x_k$ Such as 5 / 0 or 5 / 1. **7 pts**

k=5, w = (3, 4, 6, 10, 20), a = (2, 3, 4, 4, 5), t= 10

k/t	0	1	2	3	4	5	6	7	8	9	10
k=0	0	0	0	0	0	0	0	0	0	0	0
k=1	0/0	0/0	3/1	3/1	3/1	3/1	3/1	3/1	3/1	3/1	3/1
k=2	0/0	0/0	3/0	4/1	4/1	7/1	7/1	7/1	7/1	7/1	7/1
k=3	0/0	0/0	3/0	4/0	6/1	7/0	9/1	10/1	10/1	13/1	13/1
k=4	0/0	0/0	3/0	4/0	10/1	10/1	13/1	14/1	16/1	17/1	
k=5	0/0	0/0	3/0	4/0	10/0						

$F_4(10) = \max\{F_3(10), F_3(10-4)+10\} = \max\{13, 9+19\} = 19/1$
$F_5(5) = \max\{10, F4(0)+20\} = \max\{10, 20\} = 20/1$
$F_5(6) = \max\{10, F4(0)+20\} = \max\{13, 20\} = 20/1$
$F_5(7) = \max\{F_4(7), F_4(2)+20\} = \max\{14, 3+20\} = 23/1$
$F_5(8) = \max\{F_4(8), F_4(3)+20\} = \max\{16, 4+20\} = 24/1$
$F_5(9) = \max\{F_4(9), F_4(4)+20\} = \max\{17, 10+20\} = 30/1$
$F_5(10) = \max\{F_4(10), F_4(5)+20\} = \max\{19, 10+20\} = 30/1$

ii) determine the solution vector $x = (x_1, x_2, x_3, x_4, x_5)$ for t=9 and t=10 **4 pts**

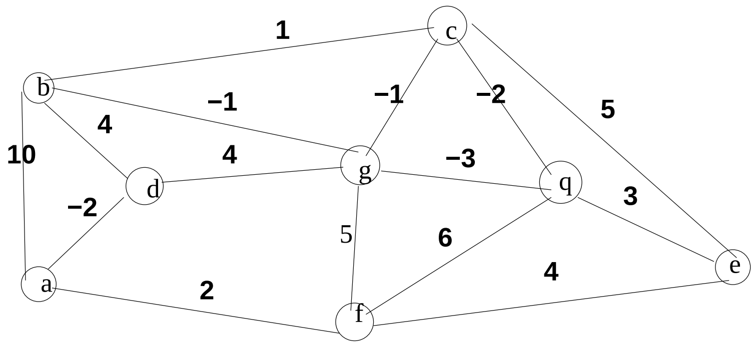
Solution:

t=9, X=(0,0,0,0,0). k=5, $p_k(t) = 1 \rightarrow x_5 = 1$ t=t-a_5=9-5=4
t=4, k=4, $p_4(4) = 1 \rightarrow x_4 = 1, t = t - a_4 = 4 - 4 = 0$. Thus X=(0,0,0,1,1).

t=10, X=(0,0,0,0,0). k=5, $p_k(t) = 1 \rightarrow x_5 = 1$ t=t-a_5=10-5=5
t=5, k=4, $p_4(5) = 1 \rightarrow x_4 = 1, t = t - a_4 = 5 - 4 = 1$. $p_3(1) = 0 \rightarrow x_3 = 0, t = 1, p_2(1) = 0, x_2 = 0, p_1(1) = 0, x_1 = 0$. Thus X=(0,0,0,1,1).

Algorithm to compute vector X:
for k=n to 1 by -1 do
let $s = p_k(t)$
$x_k = s$
$t = t - x_k * a_k$
end for

8. Given the undirected graph with edge weights, for the minimum spanning tree problem



Apply 7 iterations of Greedy/Kruskal's Algorithm. Indicate 7 edges that is processed, and indicate with + or - whether that edge will be part of optimal tree ? Draw the resulting forest ! **10 pts**

index	1	2	3	4	5	6	7
edge (i,j)							
weight							
selected +/-							

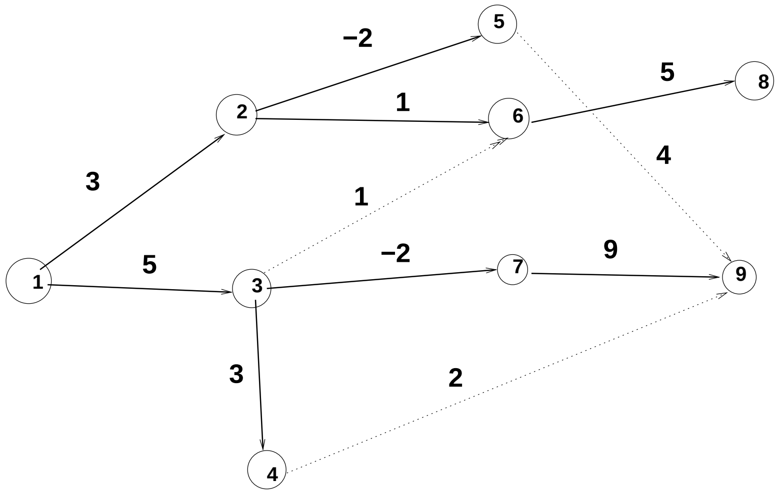
Solution:

We sort edge weight take first 7 edges and pocess them. We start with empty forest, (all nodes and no edges), and try to add the current edge to forest. If there will be a cycle we throw that edges , otherwise we add that edge to forest. A + in **selected** row means that edge is added to forest, a - otherwise.

index	1	2	3	4	5	6	7
edge (i,j)	gq	cq	ad	gc	bg	bc	af
weight	-3	-2	-2	-1	-1	1	2
selected +/-	+	+	+	-	+	-	+

Forest has two components : 1st componet is the path: (b,g), (g,q), (q,c); 2nd component is the path: (d,a), (a,f)

9. Given G as T and $T^{\perp}$ for the shortest path problem with root r=1, is T optimal? Verify it, or find an optimal tree by improving with Network Simplex Method, if there is one. Compute shortest distances and parent pointers in the final tree. **10**



k	1	2	3	4	5	6	7	8	9
d	0								
p	-								

Solution : Intial tree distances are:
d(1)=0, d(2)=3, d(5)=1, d(6)=4, d(8)=9, d(3)=5, d(7)=3, d(9)=12, d(4)=8.

Optimality condition for co-tree edge e=(i,j) is

$$d(i) + w(i, j) \geq d(j)$$

If the above condition is violated then T'= T + e -f where f=(p(j), j), p(j) parent of node j in the tree.

edge (3,6) satisfies the above condition: $d(3) + w(3, 6) = 5 + 1 = 6 > d(6) = 4$.
for edge (4,9) we have $d(4) + w(4, 9) = 8 + 2 < 12$ and for edge (5,9) we have $d(5) + w(5, 9) = 1 + 4 < 12$. We can choose (5,9) or (4,9) to pivot: (5,9) finds optimal solution in 1 itration, whereas you need to pivots if you start with (4,9).

Let us pick e=(5,9). Then f=(p(9),9)=(7,9). T'= T + e -f will be optimal.

k	1	2	3	4	5	6	7	8	9
d	0	3	5	8	1	4	3	9	5
p	-	1	1	3	2	2	3	6	5