Dat 158 Algorithms 2

587728
Joachim Leiros
Andreas Seljeset

Have during the assignment cooperated with group 23 about the work.

1.
Given 7 job with corresponding processing times

Job number	1	2	3	4	5	6	7
Processing time	4	7	2	9	12	5	3

The jobs will be executed by 3 identical machines, and all jobs are available at time 0
M=3
a) Local search algorithm

Local search Algo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	1	1	1	1	4	4	4	4	4	4	4	4	4		
M2	2	2	2	2	2	2	2	6	6	6	6	6	7	7	7
M3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	

Is done at 15

b) Greedy algorithm

Greedy	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	1	1	1	1	4	4	4	4	4	4	4	4	4		
M2	2	2	2	2	2	2	2	6	6	6	6	6	7	7	7
M3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	

Is also done at 15

c) Longest processing time rule

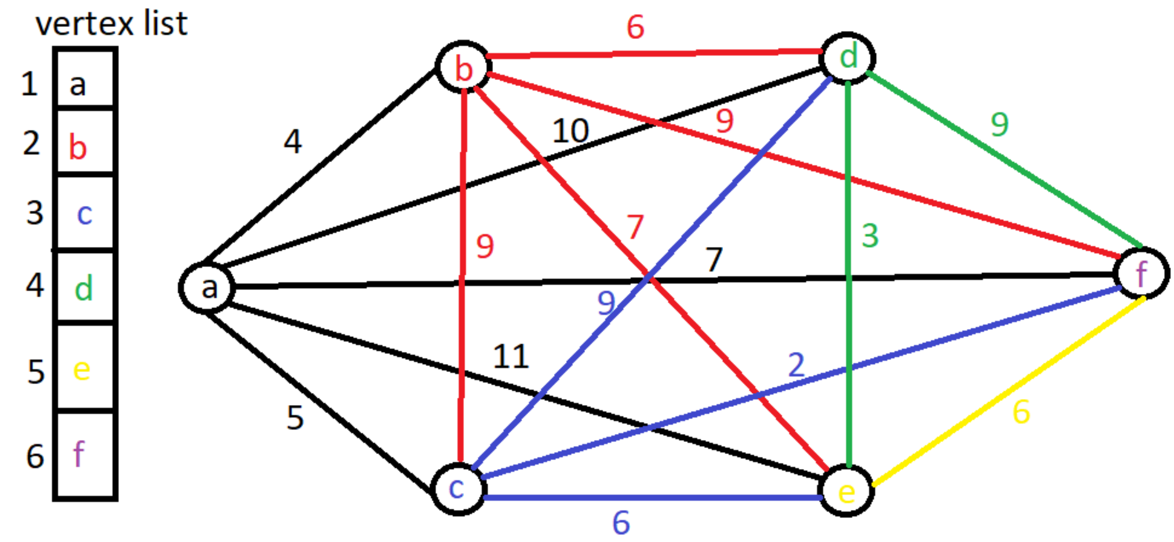
Longest PTR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	5	5	5	5	5	5	5	5	5	5	5	5	3	3	
M2	4	4	4	4	4	4	4	4	4	1	1	1	1		
M3	2	2	2	2	2	2	2	6	6	6	6	6	7	7	7

It has a complete processing time off 15

d) The lowest time the 3 machines can do the tasks is in 14. This is where the machines can change tasks freely and will always do the task with the longest time remaining.

e) It should be, but this might not be what potential users want. It might be better for the user if some tasks get done before others, instead of waiting for everything to be finished.

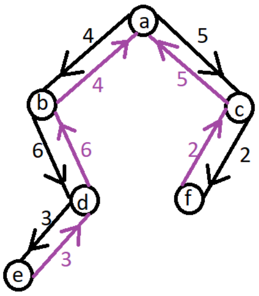
2.



a) to verify that the triangle inequality holds when going from vertex 1 to vertex 5, im gonna prove that going direct from 1 point to another will be more efficient or equal compared to when you make detours.
and example of this is going from point a to e that are showed under:
example:
a -> e:
a -> e = 11
a -> b -> e = 4 + 7 = 11
a -> c -> e = 5 + 6 = 11
a -> d -> e= 10 + 3 = 13
a -> f -> e = 7 + 6 = 13
a->e <= a ->b -> e
a->e <= a->c->e
a->e <= a->d->e
a->e <=a->f->e

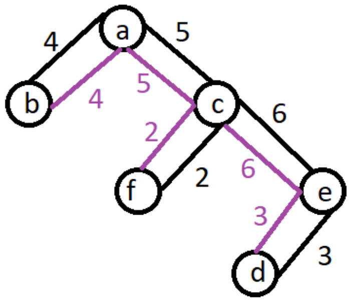
b) running the nearest addition algorithm on node. we find the two verticles that are closest together and find out that c and f are closest with the cost of 2, so we start on these:
c -> f -> e -> d -> c = 2 + 6 + 3 + 9 = 20

c) Double tree algorithm on minimum spanning tree 1 :



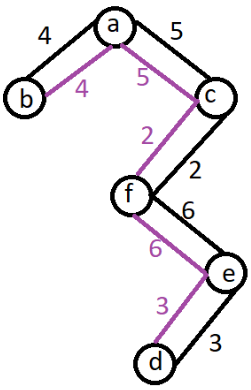
a -> b -> d -> e -> d -> b -> a -> c -> f -> c -> a
removing the already visited vertices and we get:
a ->b -> d - > e -> f -> c -> a
= 4 + 6 + 3 + 6 + 2 + 5 = 26 is the your length

Double tree algorithm on minimum spanning tree 2:



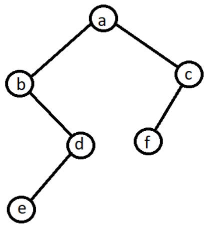
a -> b -> a ->c -> e > d -> e ->c -> f -> c -> a
removing the already visited verticles and we get:
a -> b -> c -> e -> d -> f -> a
= 4 + 9 + 6 + 3 + 9 + 7 = 38 is the tour length

Double tree algorithm on minimum spanning tree 3:



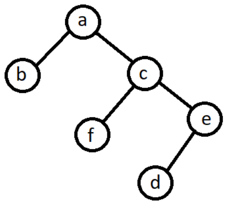
a -> b -> a -> c -> f -> e -> d -> e -> f -> c -> a
removing the already visited verticles and we get:
a -> b -> c -> f -> e -> d -> a
= 4 + 9 + 2 + 6 + 3 + 10 = 34 is the tour length

d) Minimum spanning tree 1:



In this minimum spanning tree we have 2 nodes that has an odd degree, node e and node f so we connect them to get a perfect matching euler tour:
a -> b -> d -> e -> f -> c -> a and get 4 + 6 + 3 + 6 + 2 + 5 = 26 as a tour length

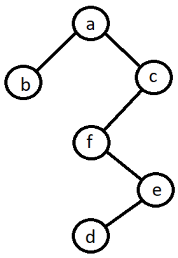
Minimum spanning tree 2:



In this minimum spanning tree we have 43 nodes that have an odd degree.
b and d are odd and has a low cost, and are therefore a good candidate for minimum perfect matching and f and c are also low, but they are already connected, but this is not a problem so we add another edge from c to f.

euler tour:
a -> b -> d -> e -> c -> f -> c -> a
we than remove the visited vertices:
a -> b -> d -> e -> c -> f -> a and get 4 + 6 + 3 + 6 + 2 + 7 = 28 as a tour length

Minimum spanning tree 3:



In this minimum spanning tree we have 2 nodes that has an odd degree, node b and d we than connect them to get a perfect matching euler tour:

a -> b -> d -> e -> f -> c -> a and get a tour length of 4 + 6 + 3 + 6 + 2 + 5 = 26 as a tour length
After having done all the Minimal spanning tree we can see that tree nr 1 and 3 has 26 as euler tour values and are the best ones.

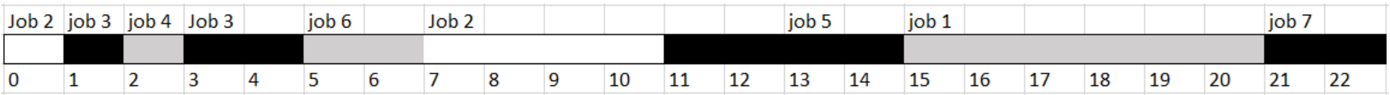
3.
a)
Item, Size, Value
1 (3,5)
2 (2,3)
3 (2,2)
4 (4,5)
5 (3,4)
6 (1,2)

B = 10

A(1)={(0,0), (3,5)}
A(2)={(0,0), (3,5), (2,3), (5,8)}
No dominant pair
A(3)={(0,0), (3,5), (2,3), (5,8), (2,2), (5,7), (4,5), (7,10)}
A(3)={(0,0), (3,5), (2,3), (5,8), (7,10)}
A(4)={(0,0), (3,5), (2,3), (5,8), (7,10), (4,5), (7,10), (6,8), (9,12), (11,15)}
A(4)={(0,0), (3,5), (2,3), (5,8), (7,10), (9,12)}
A(5)={(0,0), (3,5), (2,3), (5,8), (7,10), (9,12), (3,4), (6,9), (5,7) (8,12), (10,14), (12,16)}
A(5)={(0,0), (3,5), (2,3), (5,8), (7,10), (6,9), (8,12), (10,14)}
A(6)={(0,0), (3,5), (2,3), (5,8), (7,10), (6,9), (8,12), (10,14), (1,2), (4,7), (3,5), (6,10), (8,12), (7,11), (9,14), (11,16)}
A(6)={(0,0), (3,5), (2,3), (5,8), (8,12), (1,2), (4,7), (6,10), (7,11), (9,14)}

b)
The maximum number of elements A(i) can achieve is b+1 or 11 i
c)
Them maximum elements expressed by $V = 2^n$

4.



C1 + C2 + C3 + C4 + C5 + C6 + C7 = 2 + 4 + 6 + 10 + 14 + 20 + 22 = 78

5.

a)

```
# We make a random set
random(length)

        #create a list to fill with booleans
        boolean = {}

        #random boolean for each entry of the list
        for i in range(1,length)
                boolean['x'] = bool(random.randint(0,1))
        return boolean
```

b)

Xn = True	x1	x2	x3	x4	x5
$C1 = 3(x1 \vee -x2 \vee x3)$	3 * 1 = 3				
$C2 = 4(x1 \vee -x3)$	4 * 1 = 4				
$C3 = 1(-x1 \vee x2 \vee -x5)$	1 * ¾ = ¾	1 * 1 = 1			1 * ¾ = ¾
$C4 = 3(-x1 \vee x3)$	3 * ½ = 1 ½		3*1 = 3		
$C5 = 2(x2 \vee -x4)$		2*1 = 2		2*½ = 1	
$C6 = 7(-x2 \vee x5 \vee x9)$		7 * ¾ = 5.25			
$C7 = 2(x3 \vee -x4)$			2 * 1 = 2		

$C8 = 3(-x4 \vee -x5 \vee x21)$				3 * ¾ = 2.25	
$C9 = 5(x5 \vee -x12)$					5*1 = 5
Total	37/4	33/4	20/4	13/4	23/4

Xn = False	x1	x2	x3	x4	x5
$C1 = 3(x1 \vee -x2 \vee x3)$	3 * ¾ = 2.25				
$C2 = 4(x1 \vee -x3)$	4 * ½ = 2				
$C3 = 1(-x1 \vee x2 \vee -x5)$	1 * 1	1 * ¾ = ¾			1 * 1 = 1
$C4 = 3(-x1 \vee x3)$	3 * 1		3 * ½ = 1.5		
$C5 = 2(x2 \vee -x4)$		2 * ½ = 1		2*1 = 2	
$C6 = 7(-x2 \vee x5 \vee x9)$		7 * 1 = 7			
$C7 = 2(x3 \vee -x4)$			2 * ½ = 1		
$C8 = 3(-x4 \vee -x5 \vee x21)$				3 * 1 = 3	
$C9 = 5(x5 \vee -x12)$					5 * ½ = 2.5
Total	33/4	35/4	10/4	12/4	14/4

X1 = TRUE Since x1 = T(37/4) > x1 = F(33/4)
X2 = FALSE Since x2 = T(33/4) < x2 = F(35/4)
X3 = TRUE Since x3 = T(20/4) > x3 = F(10/4)
X4 = FALSE Since x4 = T(13/4) < x4 = F(12/4)
X5 = TRUE Since x5 = T(23/4) > x5 = F(14/4)

6.

$C_1 = 4(x_1 \vee \overline{x_3}),\ C_2 = (3(\overline{x_1} \vee x_4),\ C_3 = 2\overline{x_2} \vee x_4,\ C_4 = 1(\overline{x_2} \vee \overline{x_3} \vee x_4)$

a) Max $4z_1 + 3z_2 + 2z_1 + z_4$

Subject

$(1 - y_3) + y_1 \geq z_1 \longrightarrow y_1 - y_3 - z_1 \geq -1$

$(1 - y_1) + y_4 \geq z_2 \longrightarrow y_4 - y_1 - z_4 \geq -1$

$(1 - y_2) + y_4 \geq z_3 \longrightarrow y_4 - y_2 - z_3 \geq -1$

$(1 - y_2) + (1 - y_3) + y_4 \geq z_4 \longrightarrow y_4 - y_2 - y_3 - z_4 \geq -2$

$y_i \in \{1, 0\},\ i = 1, 2, 3, 4$

$z_j \in \{1, 0\},\ j = 1, 2, 3, 4$

6, $(1 - y_3) + y_1 \geq z_1 \rightarrow y_1 - y_3 - z_1 \geq -1$

$(1 - y_1) + y_4 \geq z_2 \rightarrow y_4 - y_1 - z_4 \geq -1$

$(1 - y_2) + y_4 \geq z_3 \rightarrow y_4 - y_2 - z_3 \geq -1$

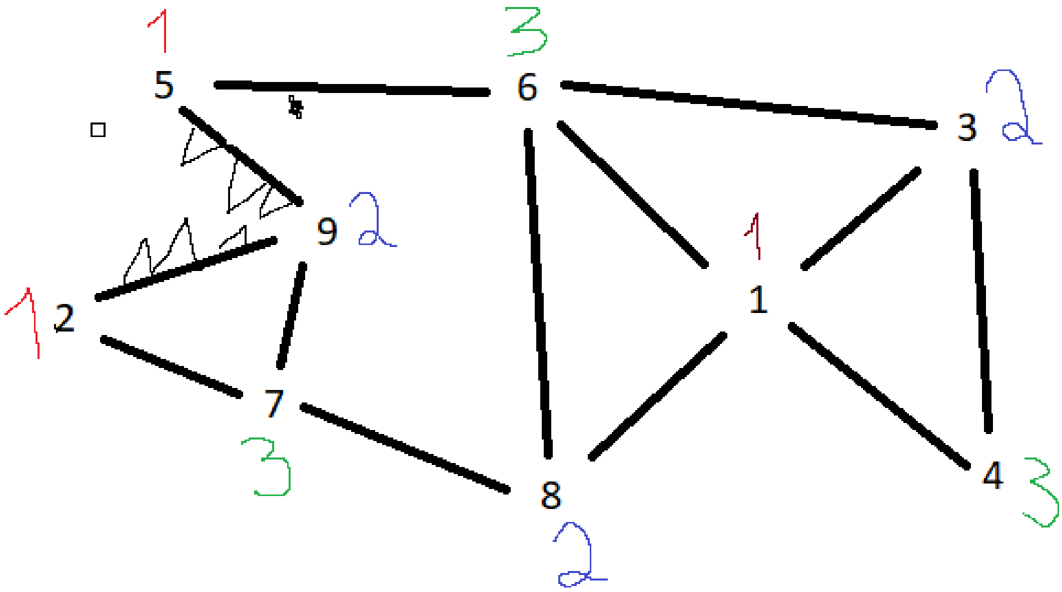
$(1 - y_2) + (1 - y_3) + y_4 \geq z_4 \rightarrow y_4 - y_2 - y_3 - z_4 \geq -2$

$0 \leq y_i \leq 1, \; i = 1, 2, 3, 4$

7.

	1	2	3	4	5	6	7	8	9
1			1	1		1		1	
2							1		1
3	1			1		1			
4	1		1						
5						1			1
6	1		1					1	
7		1						1	1
8	1					1	1		
9		1			1		1		

Node 1 goes to 3,4,6,8. Node 2 goes to 7,9. Node 3 goes to 1,4,6. Node 4 goes to 1,3 Node 5 goes to 6.9. Node 6 goes to 1,3,8. Node 7 goes to 2,8,9. Node 8 goes to 1,6,7 Node 9 goes to 2,5,7



Choose node 1 with color red, since the exercise specified to choose the lowest number, the lowest next node is three, which gets the color blue, then node four gets gets green, the next node is six, which should be getting the number 1, but since there is a node with the same number already connected we test with 2, the same problem occurred and we use 3 instead. Then we fill in node five with 1 and node eight with 2 this because of the order of nodes, and then start filling in the rest, node seven gets 3, node two get 1 and node nine get 2

8.

$$
\begin{array}{lllllll}
\text{Min} & X_1 & + X_2 & + X_3 & + X_4 & + X_5 & \\
(y_1) & X_1 & + X_2 & & & & \geq 1 \\
(y_2) & X_1 & + & X_3 & & & \geq 1 \\
(y_3) & X_1 & + & & & X_5 & \geq 1 \\
(y_4) & & X_2 & + X_3 & & & \geq 1 \\
(y_5) & & X_2 & + & & X_5 & \geq 1 \\
(y_6) & & & & X_4 & + X_5 & \geq 1
\end{array}
$$

a, ILP: $X_i \in \{0,1\}$, $X_i = \cancel{1}, 2, 3, 4, 5$

b, LP: $0 \leq X_i \leq 1 = X_i \in \mathbb{R}$

$$
\begin{array}{ll}
\text{Max} & y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \\
& y_1 + y_2 + y_3 \leq 1 \\
& y_1 + y_4 + y_5 \leq 1 \\
& y_2 + y_4 \leq 1 \\
& y_6 \leq 1 \\
& y_3 + y_5 + y_6 \leq 1
\end{array}
$$

c, $0 \leq y_i \leq$ $\quad i = 0, 1, 2, 3, 4, 5$

d, <u>Since LP = 500, OPT ILP must be ≥ 500.</u>