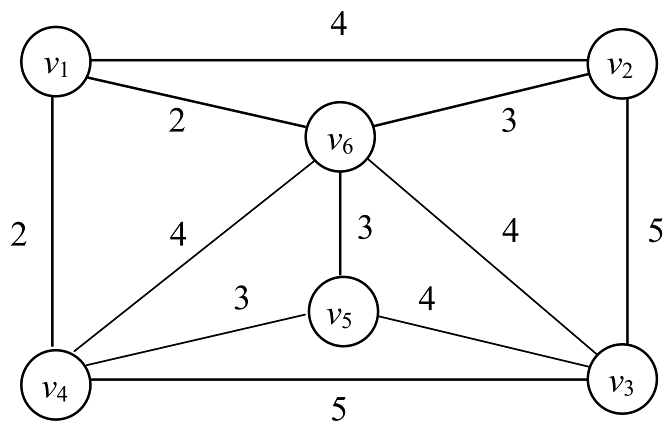
Homework 2 (計算方法設計，Design and Analysis of Algorithms)

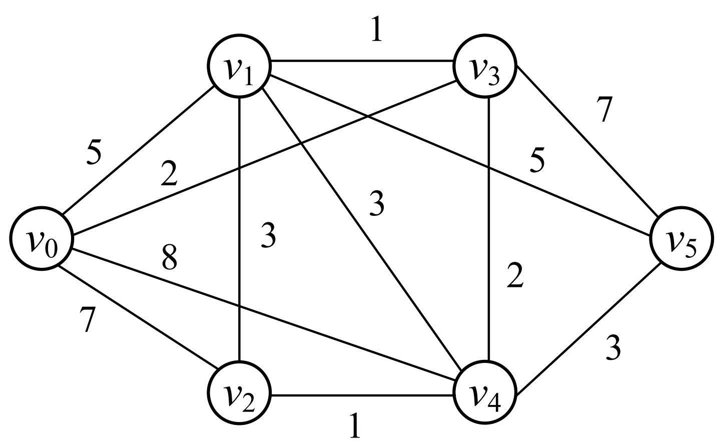
註: 所有的作業皆以紙本的方式，在截止日以前繳交給助教(台達館 737 或 738 室)，請注意不接受遲交。All homework assignments should be submitted to the TAs (Room 737 or 738 at Delta Building) as hard copy (handwriting or paper printout) by the due date. Please note that late assignment submissions will not be accepted.

Due date: April 7, 2021

1. (20%) Use (a) Kruskal's algorithm (10%) and (b) Prim's algorithm (10%) respectively to find a minimum spanning tree of the following graph.



2. (20%) Given an undirected, edge-weighted graph as shown below, use Dijkstra's algorithm to find the shortest paths from v_0 to all other nodes $v_1, v_2, \dots, v_5$.



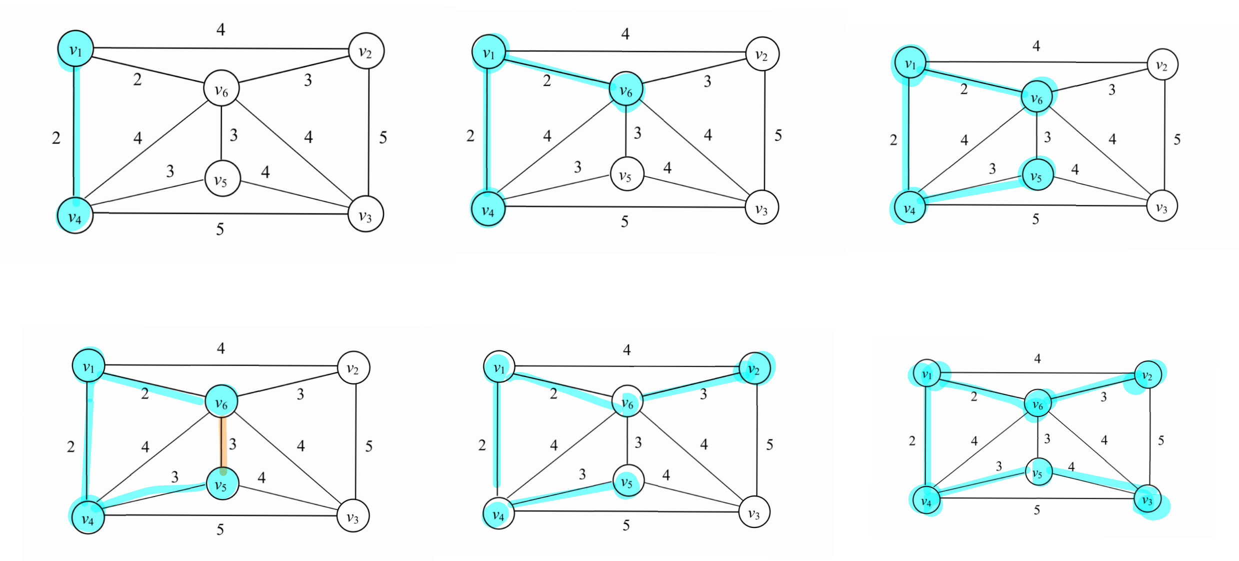
3. (20%) Modify Dijkstra's algorithm so that it can compute the shortest path from source node to each node in an arbitrary graph with negative cost edges, but no negative cycles (5%). Also prove the correctness of your algorithm (10%) and analyze the time comlexity of your algorithm (5%).

4. (20%) Encode the characters in a string message "logarithmic time method is a good algorithm" by a sequence of 0's and 1's so that the transmission cost of the encoded string message is minimum?

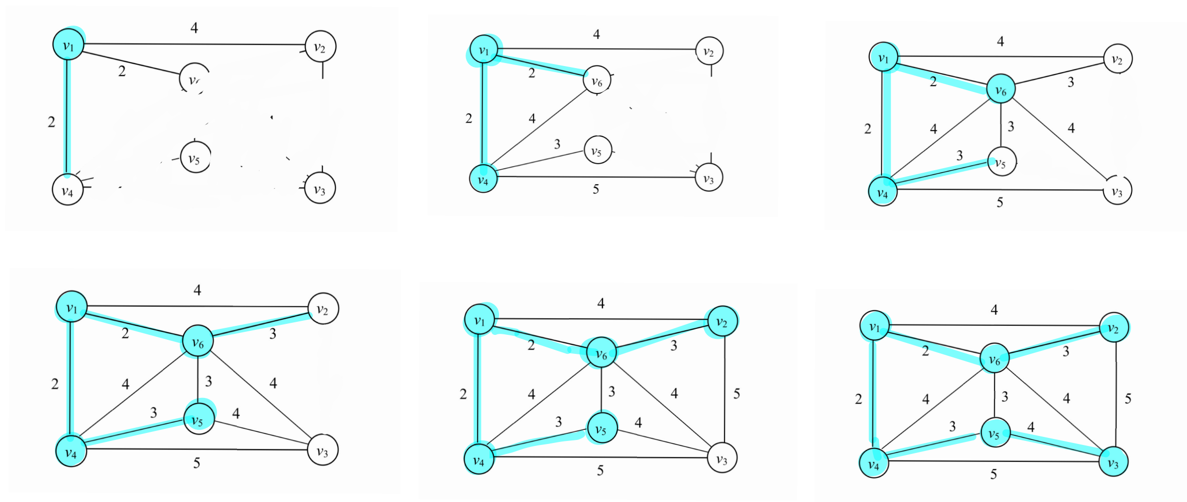
5. (20%) Let T be a set of n tasks, in which each task t_i has a start time s_i and a finish time f_i (i.e., task t_i must start at time s_i and must finish by time f_i), where $s_i < f_i$. In addition, each task has to be performed on a machine and each machine can execute only one task at a time (i.e., if two tasks overlap in time, then they cannot be scheduled to be executed on the same machine). The *task scheduling problem* is to schedule all the tasks in T such that they can be executed using the fewest machines. Please design a greedy algorithm to solve the task scheduling problem (10%). Please also prove the correctness of your greedy algorithm (10%).

1. (20%) Use (a) Kruskal's algorithm (10%) and (b) Prim's algorithm (10%) respectively to find a minimum spanning tree of the following graph.

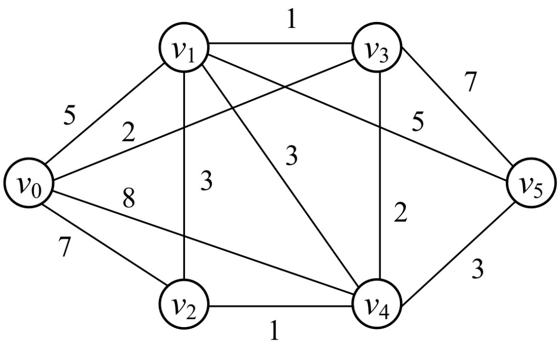
(a) Kruskal's



(b) Prim's



2. (20%) Given an undirected, edge-weighted graph as shown below, use Dijkstra's algorithm to find the shortest paths from v_0 to all other nodes $v_1, v_2, \ldots, v_5$.



S: set of marked node

$\bar{S}$: set of unmarked node

		V_0	V_1	V_2	V_3	V_4	V_5	
	d(i)	0	∞	∞	∞	∞	∞	$S = \{\emptyset\}$
	p(i)	0	0	0	0	0	0	$\bar{S}$ = 0 1 2 3 4 5
V_0	d(i)	0	5	7	2	8	∞	$S = \{0\}$
	p(i)	0	V_0	V_0	V_0	V_0	0	$\bar{S}$ = 1 2 3 4 5
V_3	d(i)	0	3	7	2	4	9	S = 0 3
	p(i)	0	V_3	V_0	V_0	V_3	V_3	$\bar{S}$ = 1 2 4 5
V_1	d(i)	0	3	6	2	4	8	S = 0 1 3
	p(i)	0	V_3	V_1	V_0	V_3	V_1	$\bar{S}$ = 2 4 5
V_4	d(i)	0	3	5	2	4	7	S = 0 1 3 4
	p(i)	0	V_3	V_4	V_0	V_3	V_4	$\bar{S}$ = 2 5
V_2	d(i)	0	3	5	2	4	7	S = 0 1 2 3 4
	p(i)	0	V_3	V_4	V_0	V_3	V_4	$\bar{S}$ = 5
V_5	d(i)	0	3	5	2	4	$\boxed{7}$	
	p(i)	0	V_3	V_4	V_0	V_3	V_4	

Shortest path

$V_0 \to V_1$: $V_0 \to V_3 \to V_1$: 3

$V_0 \to V_2$: $V_0 \to V_3 \to V_4 \to V_2$: 5

$V_0 \to V_3$: $V_0 \to V_3$: 2

$V_0 \to V_4$: $V_0 \to V_3 \to V_4$: 4

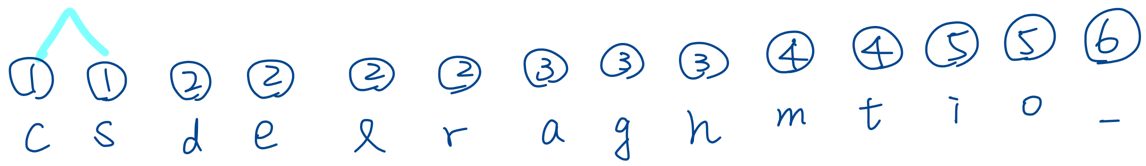
$V_0 \to V_5$: $V_0 \to V_3 \to V_4 \to V_5$: 7

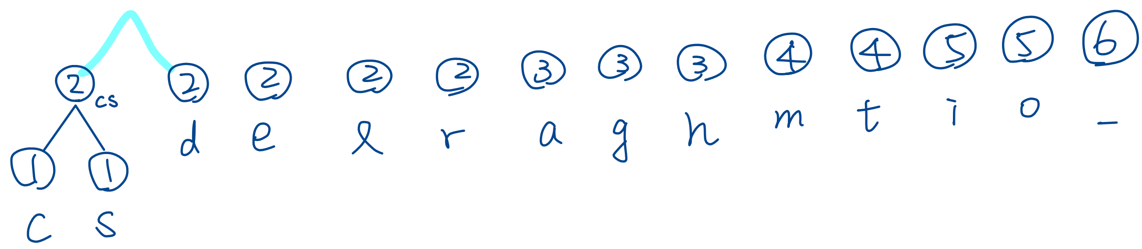
3. (20%) Modify Dijkstra's algorithm so that it can compute the shortest path from source node to each node in an arbitrary graph with negative cost edges, but no negative cycles (5%). Also prove the correctness of your algorithm (10%) and analyze the time comlexity of your algorithm (5%).

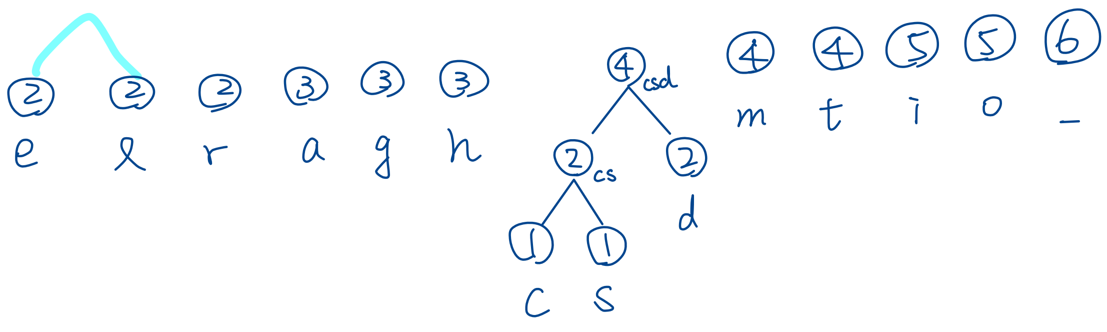
4. (20%) Encode the characters in a string message "logarithmic time method is a good algorithm" by a sequence of 0's and 1's so that the transmission cost of the encoded string message is minimum?

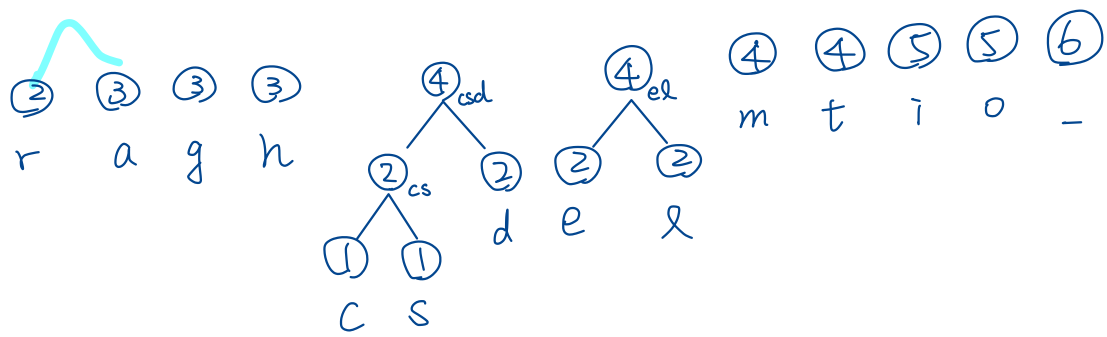
l o g a r i t h m c _ e d s

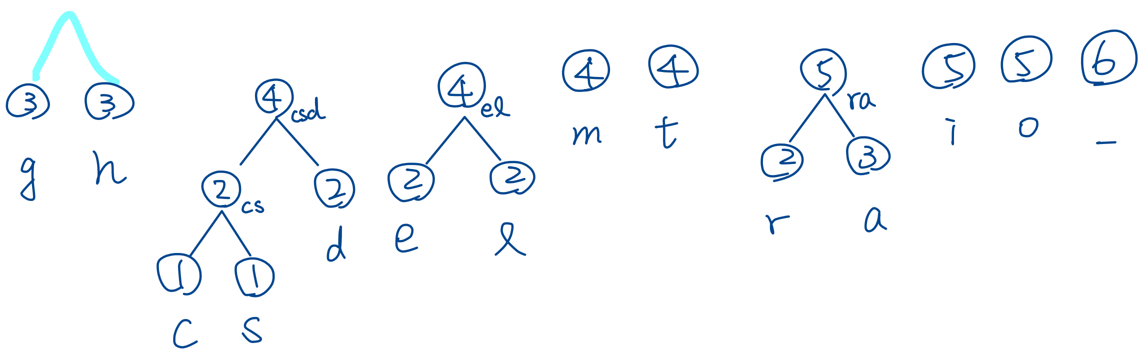
_	o	i	t	m	g	a	h	l	r	e	d	c	s
6	5	5	4	4	3	3	3	2	2	2	2	1	1



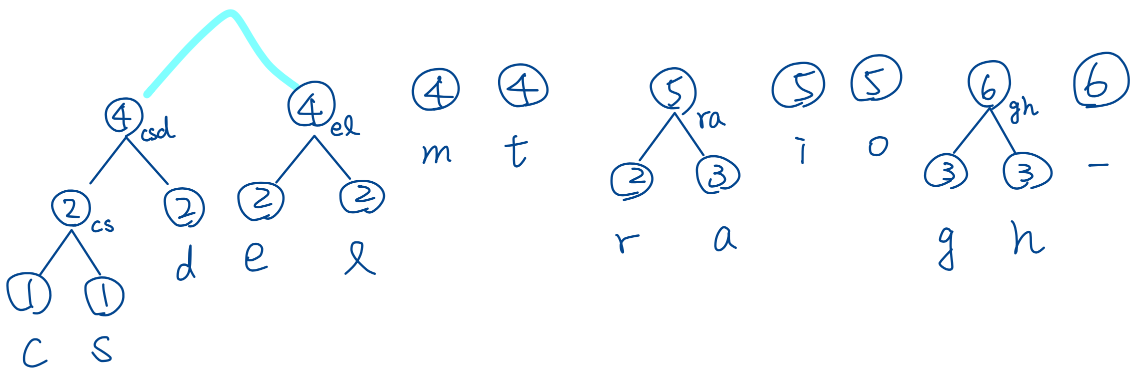




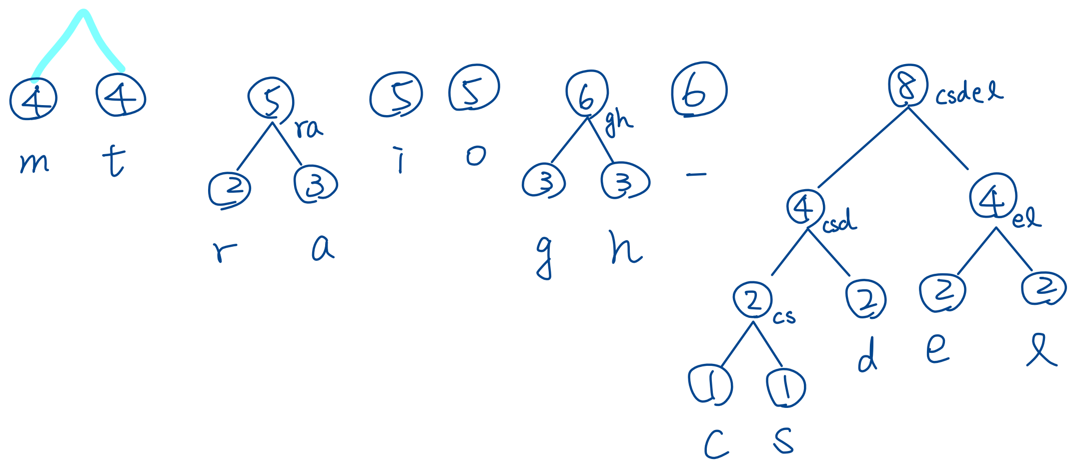




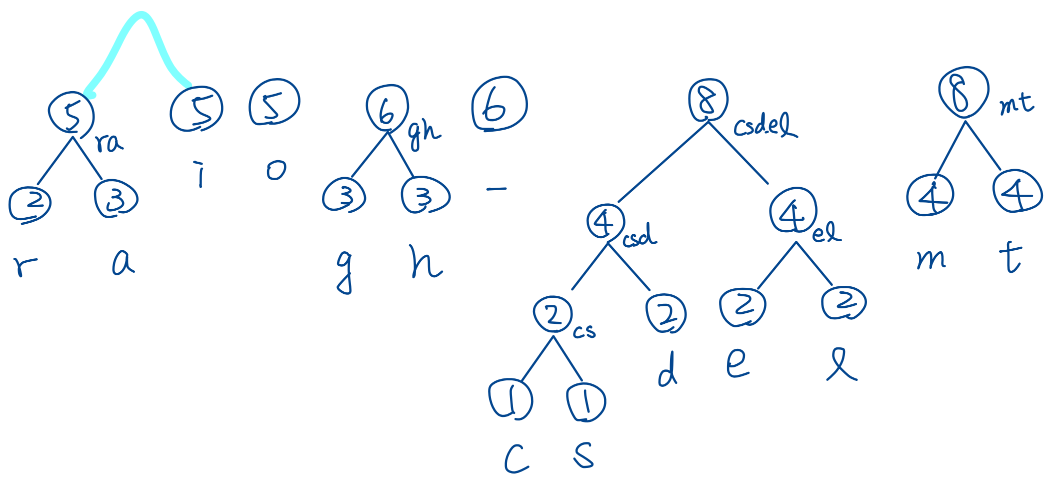
3 3
g h
4 csd
2 cs
2
1 1
c s
d
4 el
2 2
e l
4 4
m t
5 ra
2 3
r a
5 5 6
i o _



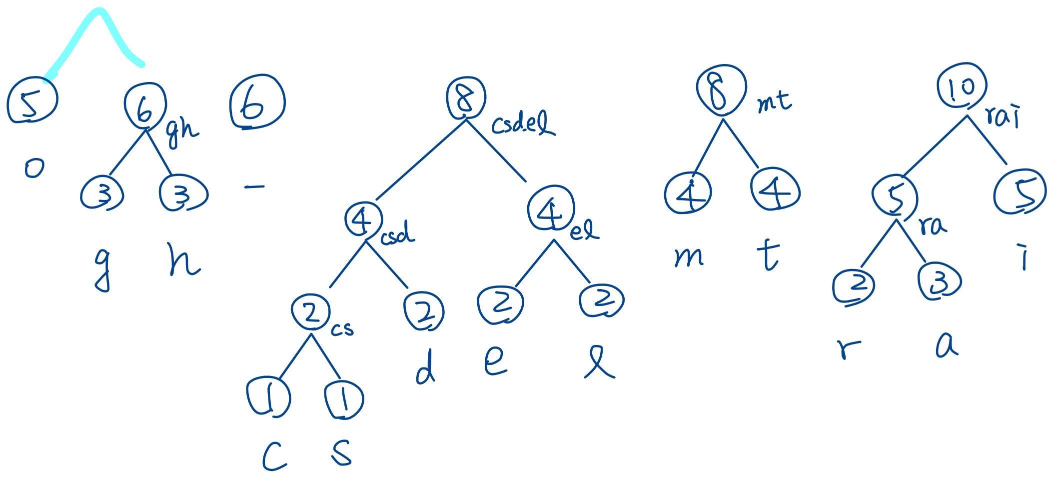
4 csd
2 cs
2
1 1
c s
d
4 el
2 2
e l
4 4
m t
5 ra
2 3
r a
5 5
i o
6 gh
3 3
g h
6
_

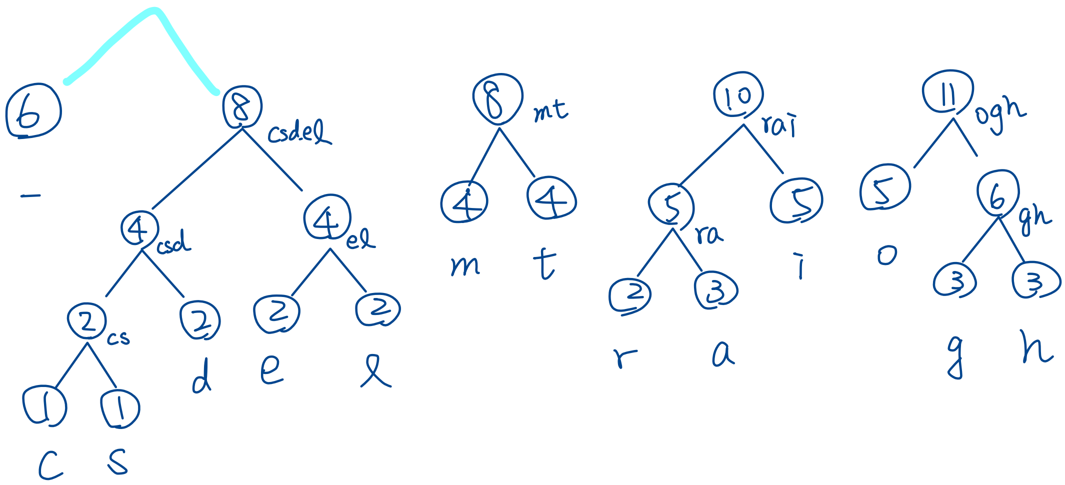


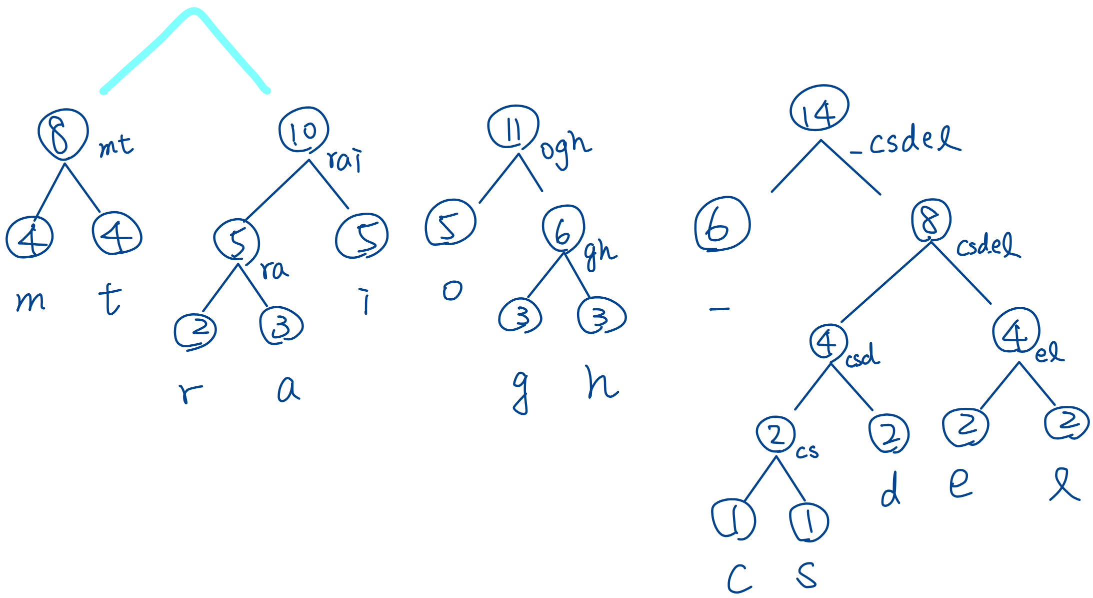
4 4
m t
5 ra
2 3
r a
5 5
i o
6 gh
3 3
g h
6
_
8 csdel
4 csd
4 el
2 cs
2
2 2
1 1
d e l
c s

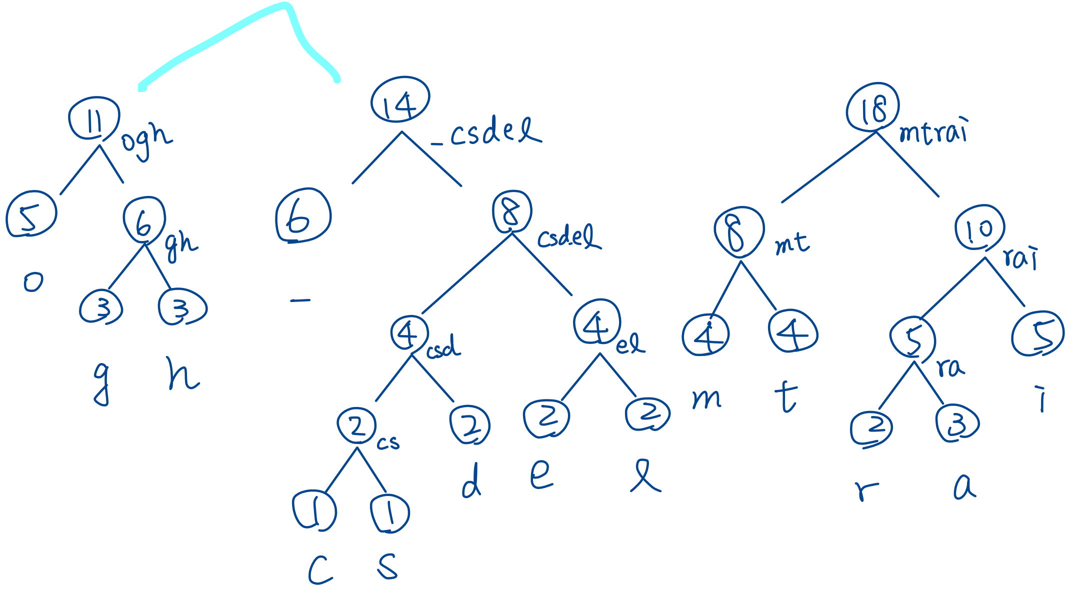


5 ra
2 3
r a
5 5
i o
6 gh
3 3
g h
6
_
8 csdel
4 csd
4 el
2 cs
2
2 2
1 1
d e l
c s
8 mt
4 4
m t

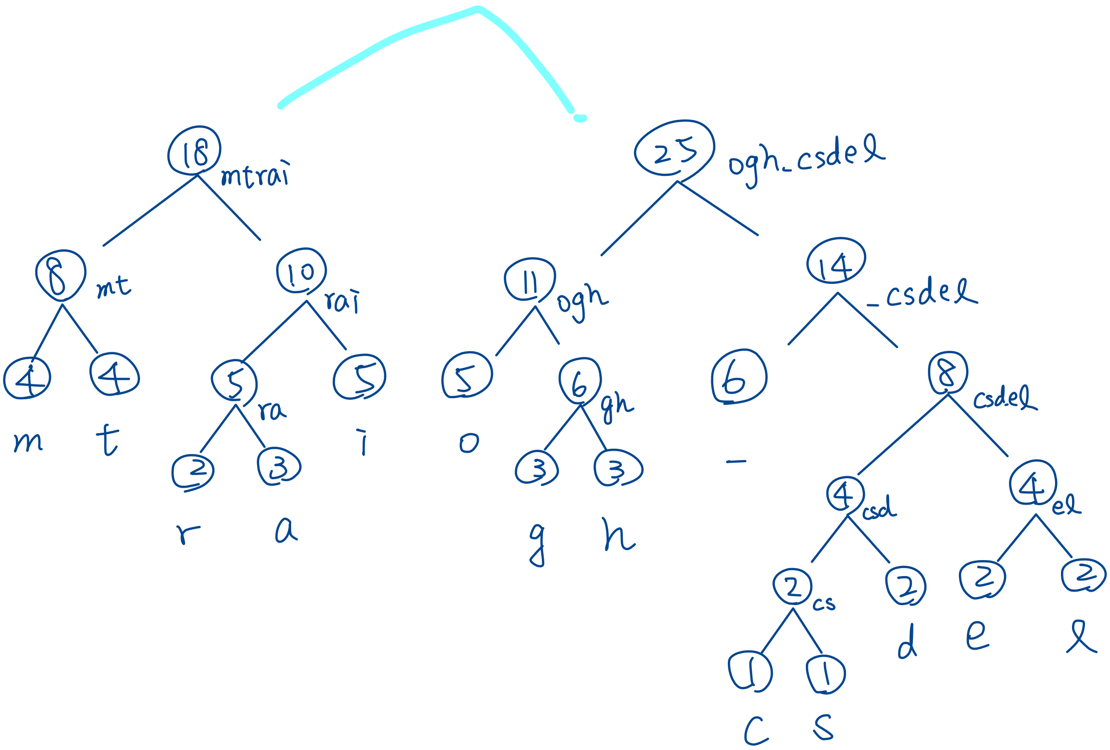

5
o
6 gh
3 3
g h
6
_
8 csdel
4 csd
4 el
2 cs
2 2 2
1 1
c s
d e l
8 mt
4 4
m t
10 rai
5 ra
5
2 3
r a
i


6
_
8 csdel
4 csd
4 el
2 cs
2 2 2
1 1
c s
d e l
8 mt
4 4
m t
10 rai
5 ra
5
2 3
r a
i
11 ogh
5
o
6 gh
3 3
g h

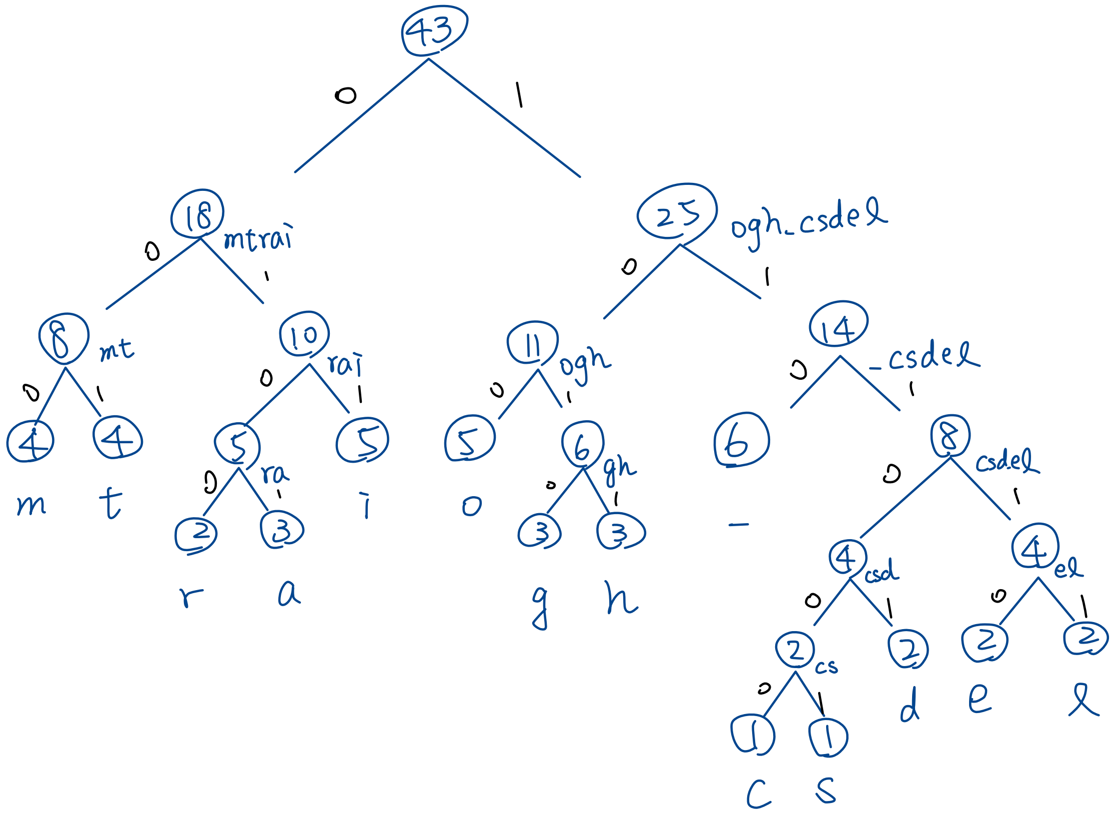

8 mt
4 4
m t
10 rai
5 ra
5
2 3
r a
i
11 ogh
5
o
6 gh
3 3
g h
14 _csdel
6
_
8 csdel
4 csd
4 el
2 cs
2 2 2
1 1
c s
d e l



11 ogh
5
o
6 gh
3
3
g
h
14 _csdel
6
_
8 csdel
4 csd
2 cs
1
1
c
s
2
d
4 el
2
2
e
l
18 mtrai
8 mt
4
4
m
t
10 rai
5 ra
2
3
r
a
5
i



18 mtrai
8 mt
4
4
m
t
10 rai
5 ra
2
3
r
a
5
i
25 ogh_csdel
11 ogh
5
o
6 gh
3
3
g
h
14 _csdel
6
_
8 csdel
4 csd
2 cs
1
1
c
s
2
d
4 el
2
2
e
l



l : 11111

o : 100

g : 1010

a : 0101

r : 0100

i : 011

t : 001

h : 1011

m : 000

c : 111000

_ : 110

e : 11110

d : 11101

s : 111001

logarithmic_

11111 100 1010 0101 0100 011 001 1011 000 011 111000 110

time_method_

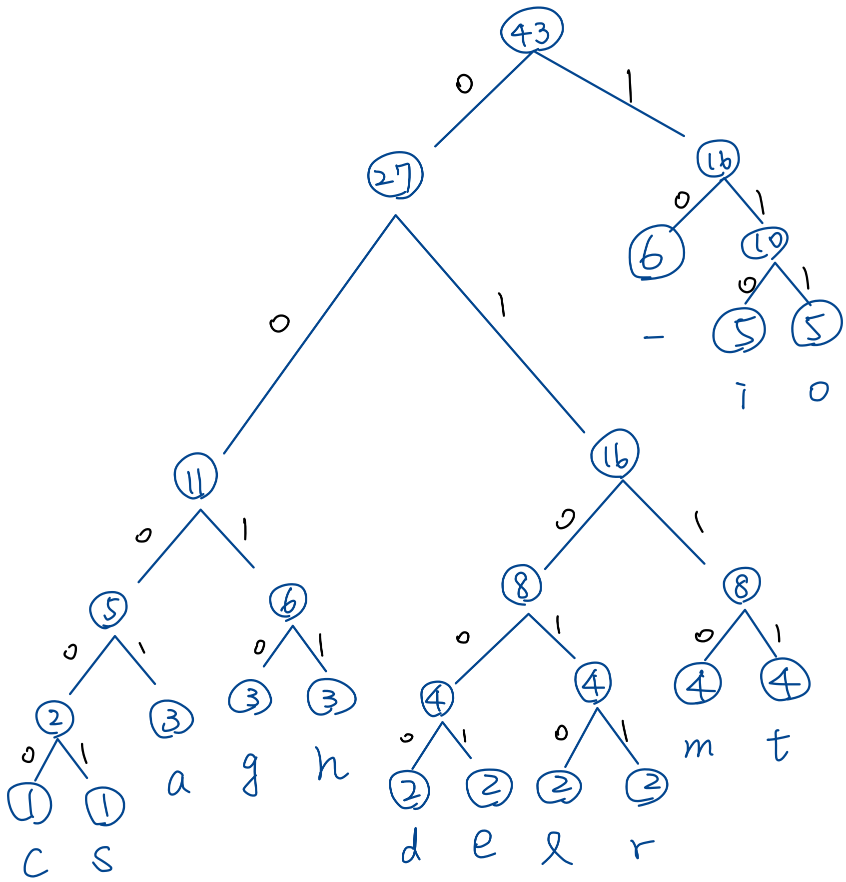
001 011 000 11110 110 000 11110 001 1011 100 11101 110

is_a_good_

011 111001 110 0101 110 1010 100 100 11101 110

algorithm

0101 11111 1010 100 0100 011 001 1011 000



l: 01010

o: 111

g: 0010

a: 0001

r: 01011

i: 110

t: 0111

h: 0011

m: 0110

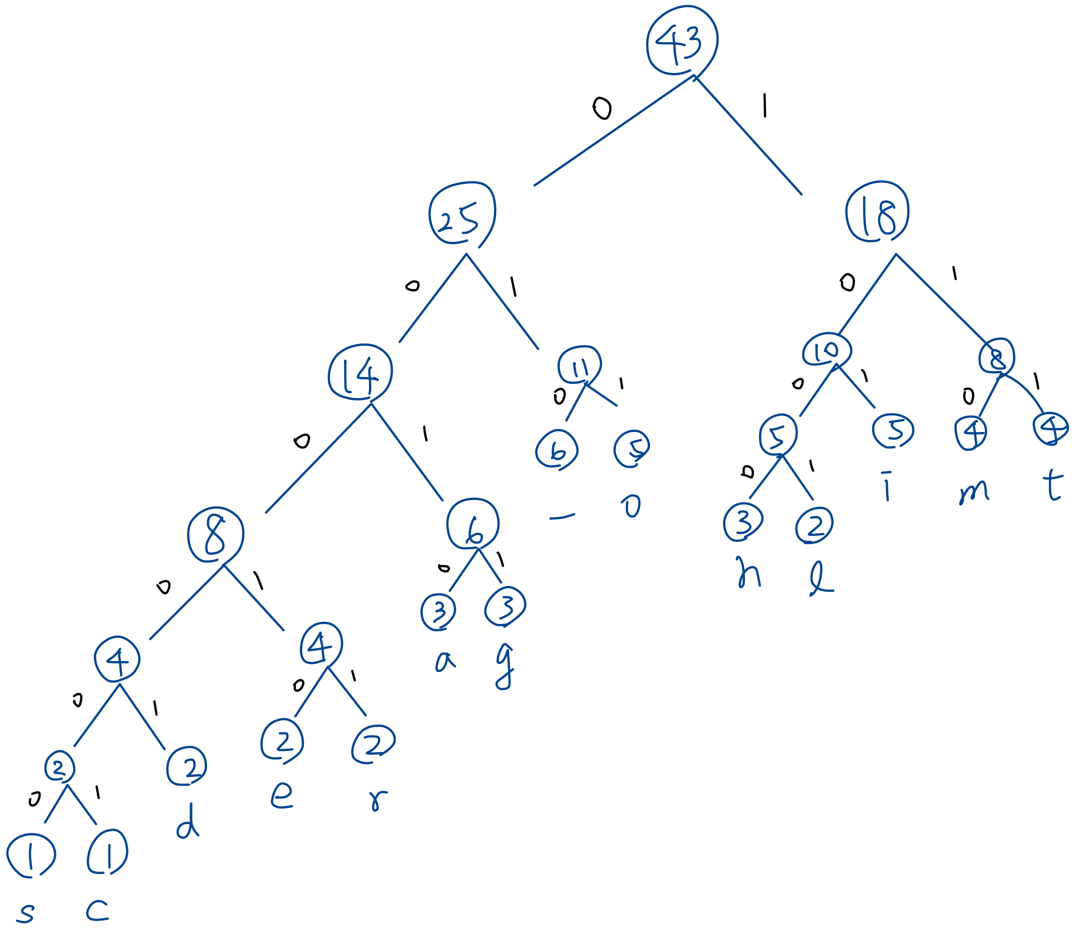
c: 00000

_: 10

e: 01001

d: 01000

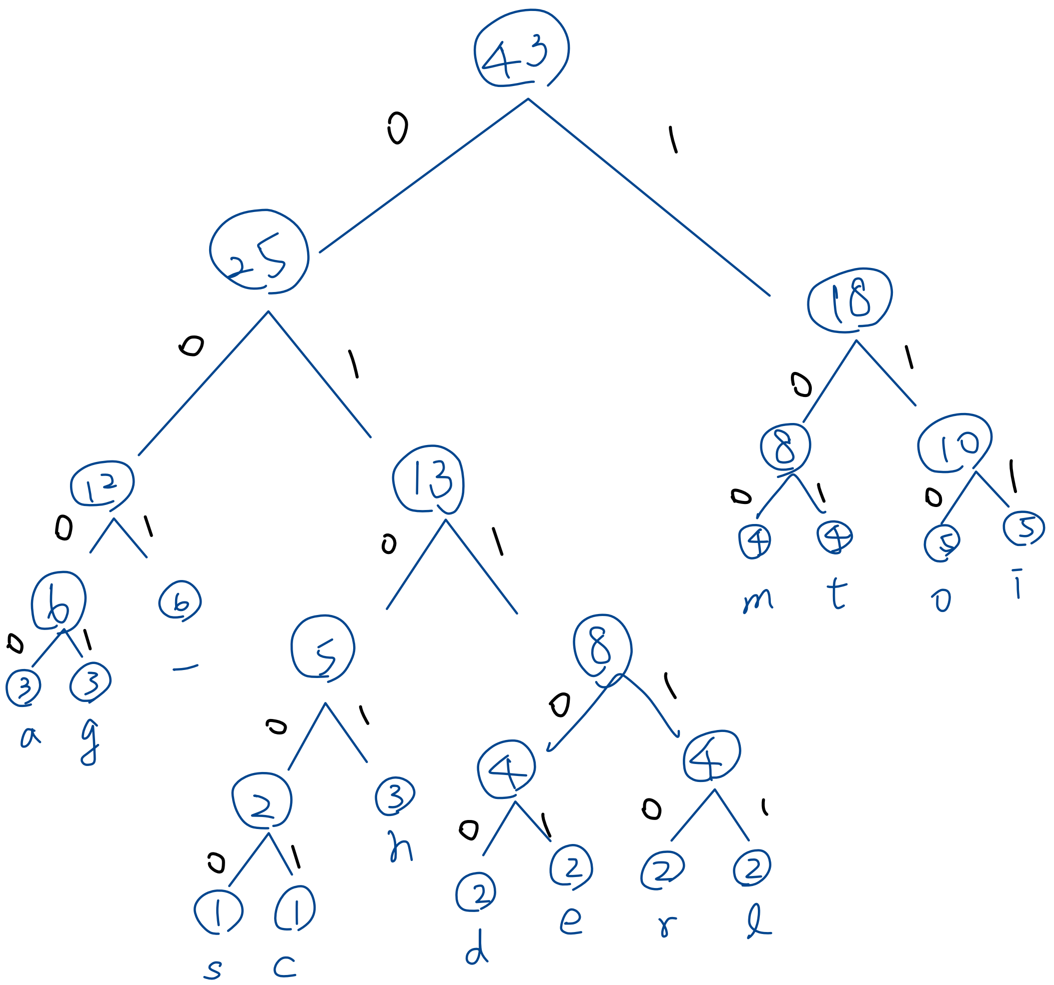
s: 00001



_ : 010
O : 011
i : 101
t : 111
m : 110

g : 0011
a : 0010
h : 1000
l : 1001
r : 00011

e : 00010
d : 00001
c : 000001
s : 000000



_ : 001
o : 110
i : 111
t : 101
m : 100

g : 0001
a : 0000
h : 0101
l : 01111
r : 01110

e : 01101
d : 01100
c : 01001
s : 01000

5. (20%) Let T be a set of n tasks, in which each task t_i has a start time s_i and a finish time f_i (i.e., task t_i must start at time s_i and must finish by time f_i), where $s_i < f_i$. In addition, each task has to be performed on a machine and each machine can execute only one task at a time (i.e., if two tasks overlap in time, then they cannot be scheduled to be executed on the same machine). The *task scheduling problem* is to schedule all the tasks in T such that they can be executed using the fewest machines. Please design a greedy algorithm to solve the task scheduling problem (10%). Please also prove the correctness of your greedy algorithm (10%).