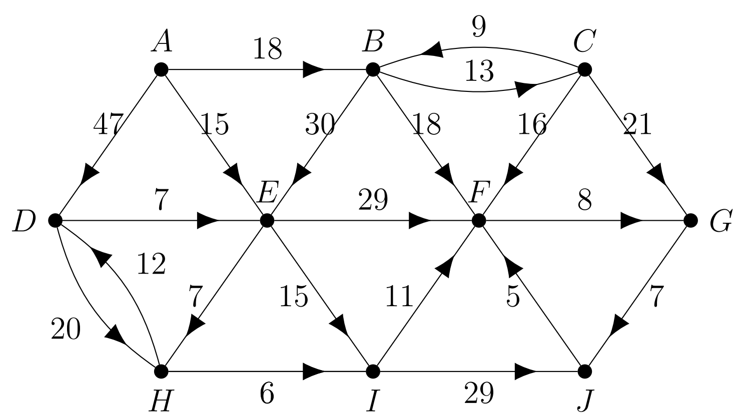
assignment 4 solutions

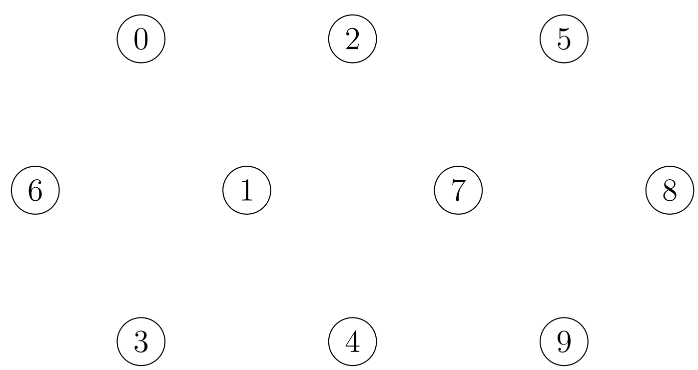
1\. (b)

j	0	1	2	3	4	5	6	7	8
L	[8	3	2	7	4	9	1	11	5]
length	[1	1	1	2	2	3	1	4	3]

(c)

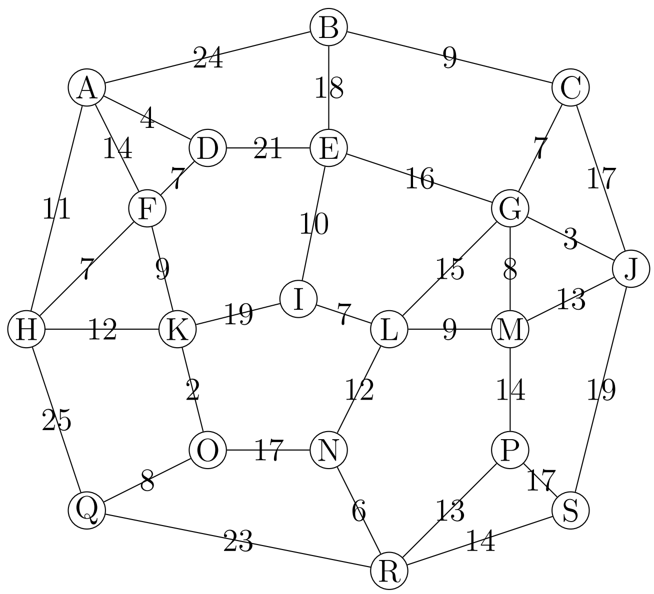
node	A	B	C	D	E	F	G	H	I	J
parent	A	A	B	H	A	B	F	E	H	G
dist	0	18	31	34	15	36	44	22	28	51





2. (a)

	-	c	l	u	b	i	n	g
-	0	1	2	3	4	5	6	7
c	1	0	1	2	3	4	5	6
a	2	1	1	2	3	4	5	6
l	3	2	1	2	3	4	5	6
b	4	3	2	2	2	3	4	5
n	5	4	3	3	3	3	3	4
o	6	5	4	4	4	4	4	4
g	7	6	5	5	5	5	5	4



(b) choices within a set can be picked in any order. KO, GJ, AD, NR, choices {CG, DF, FH, IL}, choices {GM, OQ}, choices {LM, BC, FK}, EI, LN, PR, RS, ON

(c) IL IE LM GM GJ GC CB LN NR RP RS NO OK OQ FK choices {FD, FH}, A4

(d) one. for every MST, there is some execution of Kruskal that picks it. but every execution of Kruskal produces the same tree: there are choices in the order in which edges are picked, but every choice leads to the same set of edges. so the tree found by Kruskal is the only MST.

3. (a) If e_{t+1} is in Z, then, since Z also contains S_t, Z contains $e_1 \dots e_{t+1}$, i.e. the claim holds for integer $t+1$, so we are done.

   (b) A spanning tree has exactly $n-1$ edges. If you add an edge, you have $m = n$ edges and n nodes, and any graph with $m \geq n$ must have a cycle (by Property 2 from the mst.pdf notes).

   (c) One end of e_{t+1} is in S_t; the other end is not. When we add e_{t+1} to our MST, which is a spanning tree, we create a cycle. So walk along this spanning tree from one end of e_{t+1} to the other end, and at some point you must cross from S_t to a node not in S_t.

   (d) Property 5 from the mst.pdf notes.

   (e) $w(Z') = w(Z) + w(e_{t+1}) - w(c^*) = w(Z) - (w(c^*) - w(e_{t+1}))$. So, if $w(c*) > w(e_{t+1})$, then $w(Z') <= w(Z)$, a contradiction since then Z would not be a MST.

   (f) We have shown $w(c^*) \leq w(e_{q+1})$ and $w(c^*) \geq w(e_{q+1})$. So $w(c^*) = w(e_{q+1})$, so $w(Z') = w(Z) - (w(c^*) - w(e_{t+1})) = w(Z) = 0 = w(Z)$. So Z' is a spanning tree with weight the same as a MST. So Z' is also a MST.

4. (a) Minimum number of scalar mults is upper right entry, so 42.

```
[0, 60, 22, 42]      [ -   AB   A(BC) (ABC)D ]
[-,  0, 12, 20]      [ -          BC    (BC)D ]
[-,  -,  0, 24]      [ -                  CD  ]
[-,  -,  -,  0]      [ -                      ]
```

```
entry AC: min of (AB)C, A(BC): A(BC)

entry BD: min of B(CD), (BC)D: (BC)D

entry AD: min of (ABC)D, (AB)(CD), A(BCD): (ABC)D = (A(BC))D
```

(b)

```
    *  1   2   3   4   5
20 [0, 7, 13, 17, 17, 22]
21 [0, 7, 13, 17, 17, 26]
22 [0, 7, 13, 17, 19, 26]
23 [0, 7, 13, 23, 23, 26]
```

`K[22][4]` is the max of

`K[22][3]=17`

(the best value with weight 23 using a subset of items {1,2,3})

and

`6 + K[22-9][3] = 6 + K[13][3] = 6 + 13 = 19`

(value of item 4 plus the best value with weight 22 minus weight of item 4 using a subset of items {1,2,3}),

so `K[22][4] = max( 17, 19 ) = 19 .`