CS251 Homework 4 Liyao (Mars) Gao

1.

a) Cycle DFS. O(V+E)
Starting from the vertex, say A, perform DFS until the same vertex, A was accessed.

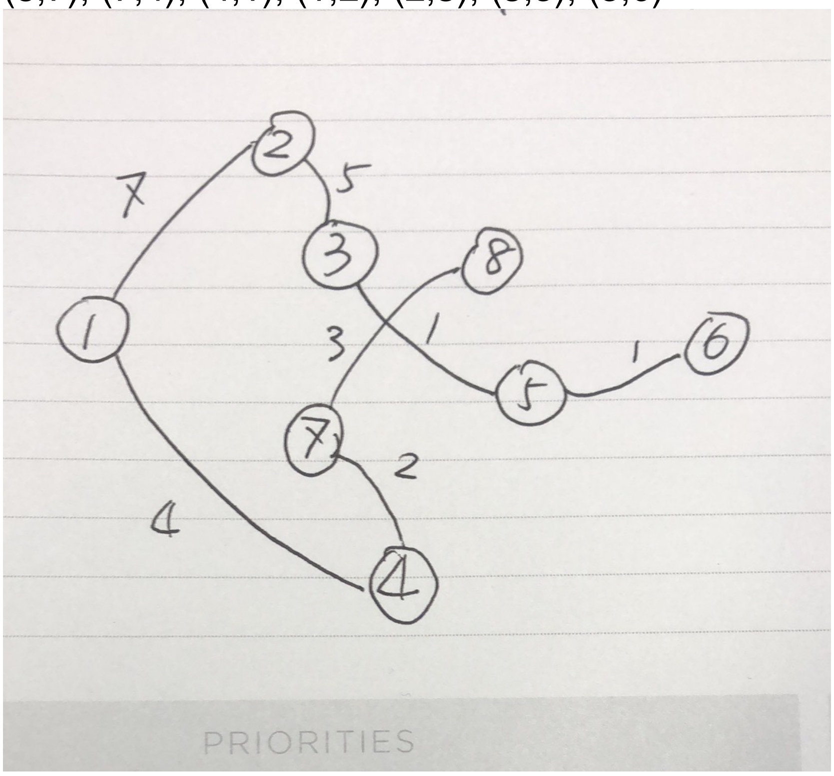
b) Floyd Warshall. O(V^3)
Fill in the adjacency table. Iterate (V-1) times for value update in the adjancency tbale.

c) Bellman-Ford. O(VE)
Iteration i finds all shortest paths that use i edges. Pick one edge each time. Update the table with the shorter path.

d) Baruvka's algorithm. O(ElogV)
Form the MST sequentially using the shortest weight edge in graph with no repletion in visited vertices list.

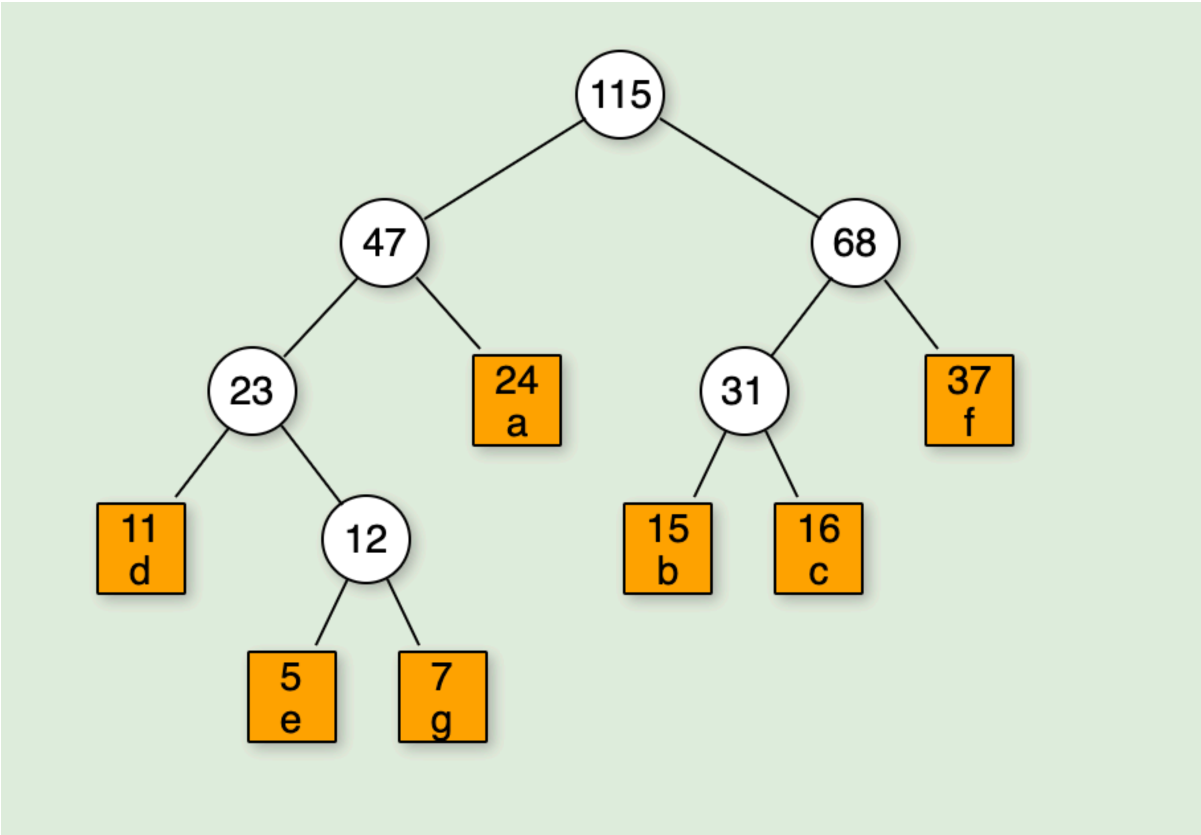
2.

(8,7), (7,4), (4,1), (1,2), (2,3), (3,5), (5,6)



Order: [8,7,4,1,2,3,5,6]

3.



4.

Suppose a minimum spanning tree of G is T.

i) If u is v's parent, grandparent, branch. If u, v are not adjacent in T.

There exist a subset of vertices, a = {a1, a2, …, an}, |a| < number of vertices in T, such that [u->a1->a2->…->v].

The sum of these edges are w(u, a1) + w(a1, a2) + … + w(an->v).

As an alternative method to connect, we can make [u->v->an->…->a1].

The sum of these edges are w(u, v) + w(v, an) + … + w(a2->a1)
= w(u, v) + w(a1, a2) + … + w(an->v)
Since w(u, v) is the smallest edge,
w(u, v) + w(v, an) + … + w(a2->a1) < w(u, a1) + w(a1, a2) + … + w(an->v).

T is not MST. Contradiction.

ii) If v is u's parent, grandparent, branch. If u, v are not adjacent in T.
For the same reason, contradiction.

iii) If u, v, have same parent, grandparent, … Suppose their parent is pu, pv.

Consider only (pu, u), (pv, v).

Sum1 = w(pu, u) + w(pv, v) + K, where K is the weights of other edges.

If we change the edges to (pu, u), (u, v), where other edges remain the same,

Sum2 = w(pu, u) + w(u, v) + K.
Since w(u, v) is the smallest edge, Sum2 < Sum1.

T is not MST. Contradiction.

5.

When length = 4,

1234 has no 341. 0 occurrence.

1234	1234
341	341
fail	fail

Fail: 2, Success: 0.

When length = 8,

12341234 has 1 occurrence.

12341234	12341234	12341234	12341234	12341234	12341234
341	341	341	341	341	341
fail	fail	success	fail	fail	fail

Fail: 5, Success: 3.

|Success| = (length - 4)/4 * 3

So when length = 1000,

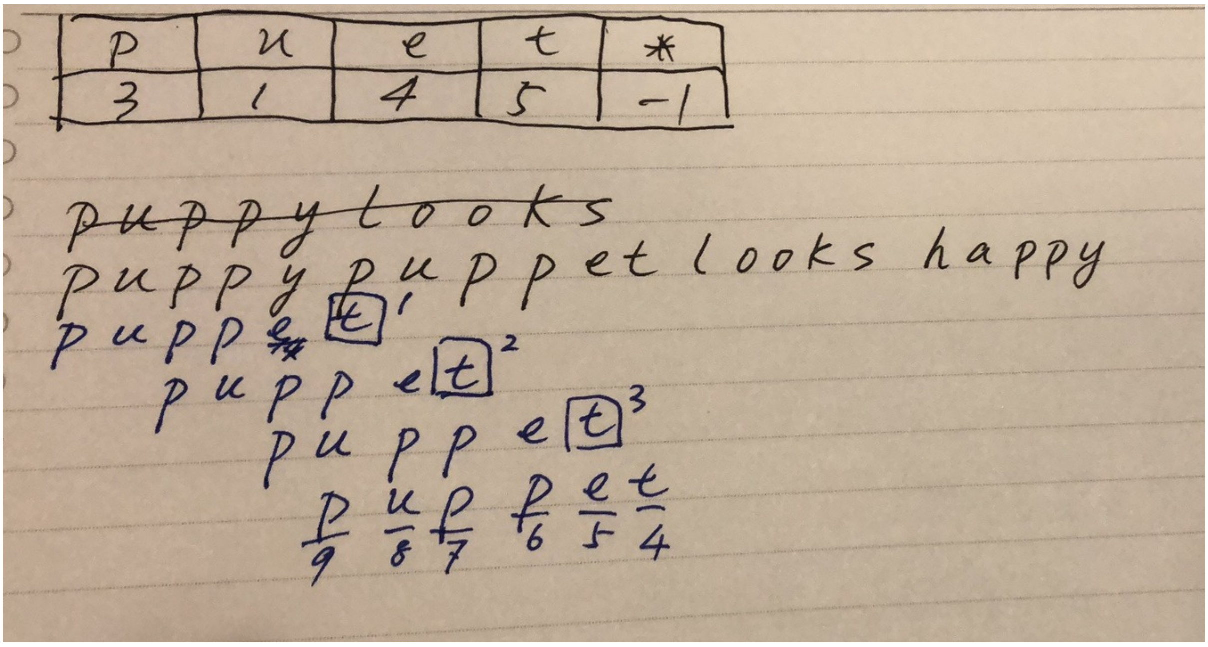
|Success| = 747
|Fail| = 749

6.

```
P: pattern, m := size of P, S := set of symbols, |S| := number of symbols
n := |S|
i := m - 1
j := n - 1
while (j != -1)
if i == -1
    S[j] := -1
        i := m - 1
        j := n - 1
else if P[i] == S[j]
    S[j] := i
        i := m - 1
        j := n - 1
else
    i := i - 1
return 0
```

7.

9 comparisons in total.



8.

```
Input: string S with n characters and P with m characters

i := m - 1
while (i != n - 1)
  if S[i] = P[m-1]
    j := 0
    while (j < m and S[i-j] == P[m-1-j])
      j := j + 1
    if j == m
      return i
    i := i + 1
return -1
```

Worst case and others: $O(n)$

9.

Answer: The output should always be a minimum spanning tree. If an edge is not in the minimum spanning tree, there should exist a smaller edge in the minimum spanning tree. The weight should be started from the biggest to the smallest. The left edges could form a MST, which fulfills the cycle property. In the end, we will have a correct tree.

10.

	0	1	2	3	4	5	6	7	8
P[j]	G	A	C	A	G	A	T	G	A
F	0	0	0	0	1	2	0	1	2

```
G G T A C C C G A C A G A T G A C A G A
G A C A G A T G A                          2
  G A C A G A T G A                        2
    G A C A G A T G A                      1
      G A C A G A T G A                    1
        G A C A G A T G A                  1
          G A C A G A T G A                1
            G A C A G A T G A
              G A C A G A T G A            9
```

2+2+1+1+1+1+1+9 = 18 in total