CS 381
HW 7
Tian Qiu

1.
(a)

1. Construct current residual network // O (E)
2. Use Edmonds Karp algorithm to construct max flow and final residual network // O(VE^2)
3. Do DFS on residual network, make reachable vertices from source // O(V+E)
4. A minimum cut consists of all edges from reachable vertex to non-reachable vertex. // O (E)

Time: Edmonds Karp algorithm dominates the time.
O(VE^2)

(b)

1. Construct current residual network // O (E)
2. Use Edmonds Karp algorithm to construct max flow and final residual network // O(VE^2)
3. Do DFS on residual network, make reachable vertices from source // O(V+E)
4. A minimum cut C consists of all edges from reachable vertex to non-reachable vertex. // O (E)
5. For all edges (u_i, v_i) in G // O (E)
    If (u_i, v_i) does not exist in residual network && minimum cut C
        // New saturated path from source to sink
        Print first minimum cut which includes (u_i, v_i), second minimum cut C
   Print the only minimum cut C // O(1)

Time: Edmonds Karp algorithm dominates the time.
O(VE^2)

2.

Given:

S: set of n points

Input line: y = mx + b

Distance from point to line:

$$\text{distance}(ax+by+c=0,(x_0,y_0)) = \frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}}$$

Pseudocode:

Initial array Distance []

// determine the point is in left or right side of y = mx + b

For each point (x_i, y_i) // O(n)

If $mx_i + b$ > y_i

Distance[i] = -1 // left side

If $mx_i + b$ = y_i

Distance[i] = 0

Else

Distance[i] = 1 // right side

For each point (x_i, y_i)

Distance[i] = Distance[i] * Distance from point to line ((x_i, y_i) , y = mx + b)

Find the median k of Distance array // O (n), use the method that introduced in class to // separate array in group of five entries to find // median of an array in O(n) time

Take line y = mx + n with (x_k, y_k) to get the number of n // O (1)

Return y = mx + n

Time: The algorithm to find the median of Distance array takes O (n) time dominating the whole algorithm.

So, time is O(n)