Ramaneek Gill 1000005754
CSC373 Problem Set 5

a) $FindMinCut\_MinS(G, s, t)$:

$Ford - Fulkerson(G, s, t)$ #first make N have max $|f|$
$S \leftarrow \emptyset$; $T \leftarrow \emptyset$
$S \leftarrow S \cup \{s\}$; $T \leftarrow V - S$
For every edge $(u, v) \in G.E$:
#can get the flow of an edge with $.f$ just like in the textbook
#first move forward edges from T to S that aren't fully used
If $u \in S \wedge v \in T \wedge (u, v).f < c_f(u, v)$:
$S \leftarrow S \cup \{v\}$; $T \leftarrow T - \{v\}$

#now move backward edges from T to S that are used
If $u \in T \wedge v \in S \wedge (u, v).f > 0$:
$S \leftarrow S \cup \{u\}$; $T \leftarrow T - \{u\}$

Return (S, T)

This algorithm finds a minimum cut with minimum $|S|$ in $O(E)$ because we are only analysing edge exactly once and since $Ford - Fulkerson()$ only takes $O(E)$ time.

b) $FindMinCut\_MinT(G, s, t)$:

$S \leftarrow \emptyset$; $T \leftarrow \emptyset$
$S \leftarrow V - T$; $T \leftarrow T \cup \{t\}$
For every edge $(u, v) \in G.E$:
#can get the flow of an edge with $.f$ just like in the textbook
#first move forward edges from S to T that aren't fully used
If $u \in S \wedge v \in T \wedge (u, v).f < c_f(u, v)$:
$T \leftarrow T \cup \{v\}$; $S \leftarrow S - \{v\}$

#now move backward edges from S to T that are used
If $u \in T \wedge v \in S \wedge (u, v).f > 0$:
$T \leftarrow T \cup \{u\}$; $S \leftarrow S - \{u\}$

Return (S, T)

This algorithm finds a minimum cut with minimum$|T|$, this is essentially a mirrored version of the first algorithm, first T starts off as small as possible and only vertices that satisfy the first two if conditions are transferred over from S to T. This keeps unnecessary vertices out of T unlike the first algorithm.