Yongzuan Wu
wu68
cs573
HW 5
Problem 3

We reduce the problem to a maximum flow problem with vertex capacity. Let $voter[1..n]$ be the set of voters, and $cand[1..m]$ be the set of candidates, and $dona[1..n][1..m]$ specifies the amount of money that voter i is willing to donate to candidate j. Let $fund[1..m]$ be the amount of money that each candidate needs to win the election. And the election lasts for T days.

To construct the flow next work, we introduces three set of vertices.

$V_d = \{d_{ij}\,|1 \leq i \leq n, 1 \leq j \leq T\}$, d_{ij} represents donation of voter i on the jth day.

$V_c = \{c_{ij}\,|1 \leq i \leq n, 1 \leq j \leq m\}$, c_{ij} represents donation from voter i to candidate j

$C = \{r_i\,|1 \leq i \leq m\}$, r_i represents candidate i.

Let $G = (V_d \cup V_c \cup C \cup \{s, t\}, E, c)$ be a flow network,

$$c(u \to v) = \begin{cases} 100 & \text{if } u = s \text{ and } v \in v_d \\ \infty & \text{if } u \in v_d \text{ and } v \in V_c \\ dona[i][j] & \text{if } u = c_{ij} \text{ and } v = r_j \\ fund[i] & \text{if } u = r_i \text{ and } v = t \\ 0 & \text{otherwise} \end{cases}$$

A flow in this network represents the actual amount of donation. It obeys the federal law since the amount of donation of every voter on each day is bounded by the capacity 100. It agrees with each voter's expected contribution since the amount of donation from voter i to candidate j is bounded by the capacity $dona[i][j]$. Thus a flow in this network is a valid donation schedule. Then the algorithm follows,

```
DonationSchedule($voter[1..n]$, $cand[1..m]$, $dona[1..n][1..m]$, $fund[1..m]$, $T$)
 construct the flow network $G$ as described above
 compute maximum flow $F$ in $G$
 if ( every edge going into $t$ is saturated)
   $schedule[i][j][k] = f(d_{ik}, c_{ij})$
 else return ('no such schedule exist')
```

The returned array $schedule[i][j][k]$ reports the amount of money voter i donates to candidate j on the kth day.

Analysis: Computing the maximum flow with Dinitz algorithm with dynamic trees costs $O(VElog(V)) = O((nm + nT)nmTlog((nm + nT)))$. By construction, $|V| = O(nm + nT)$, $|E| = O(nmT)$. The cost to constructing the network is relatively small, thus obsorbed in the big O notation. Therefore the running time is $O((nm + nT)nmTlog(nm + nT))$.