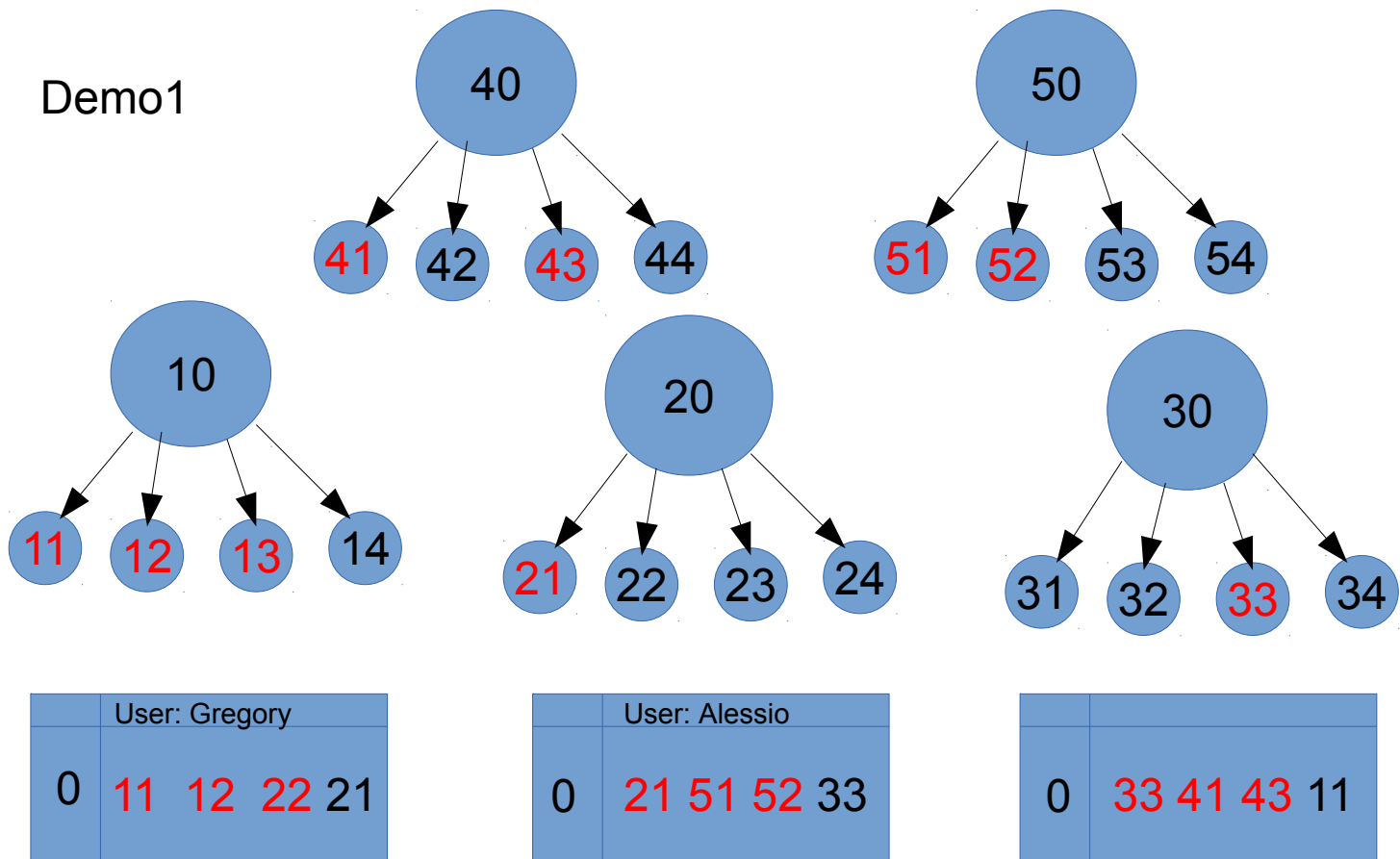


Demo1

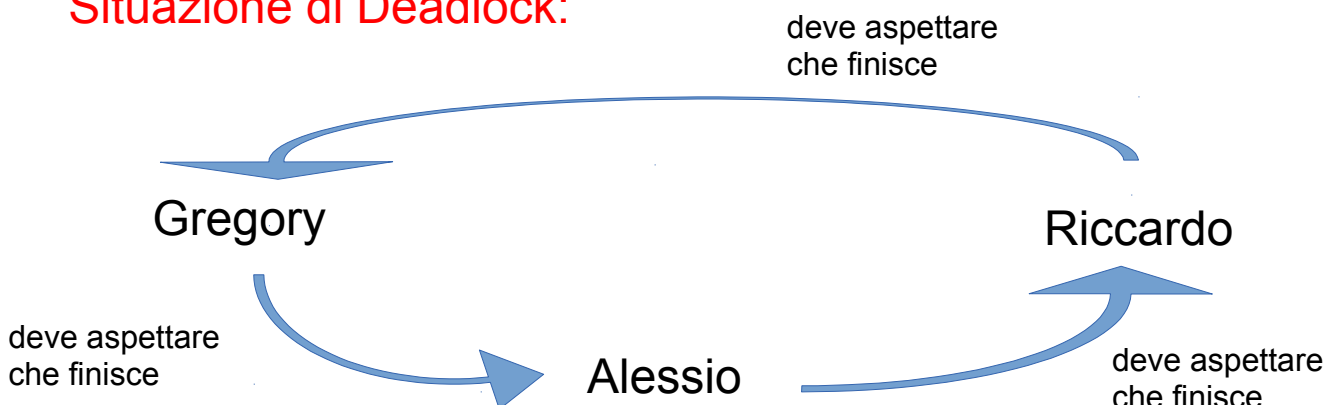


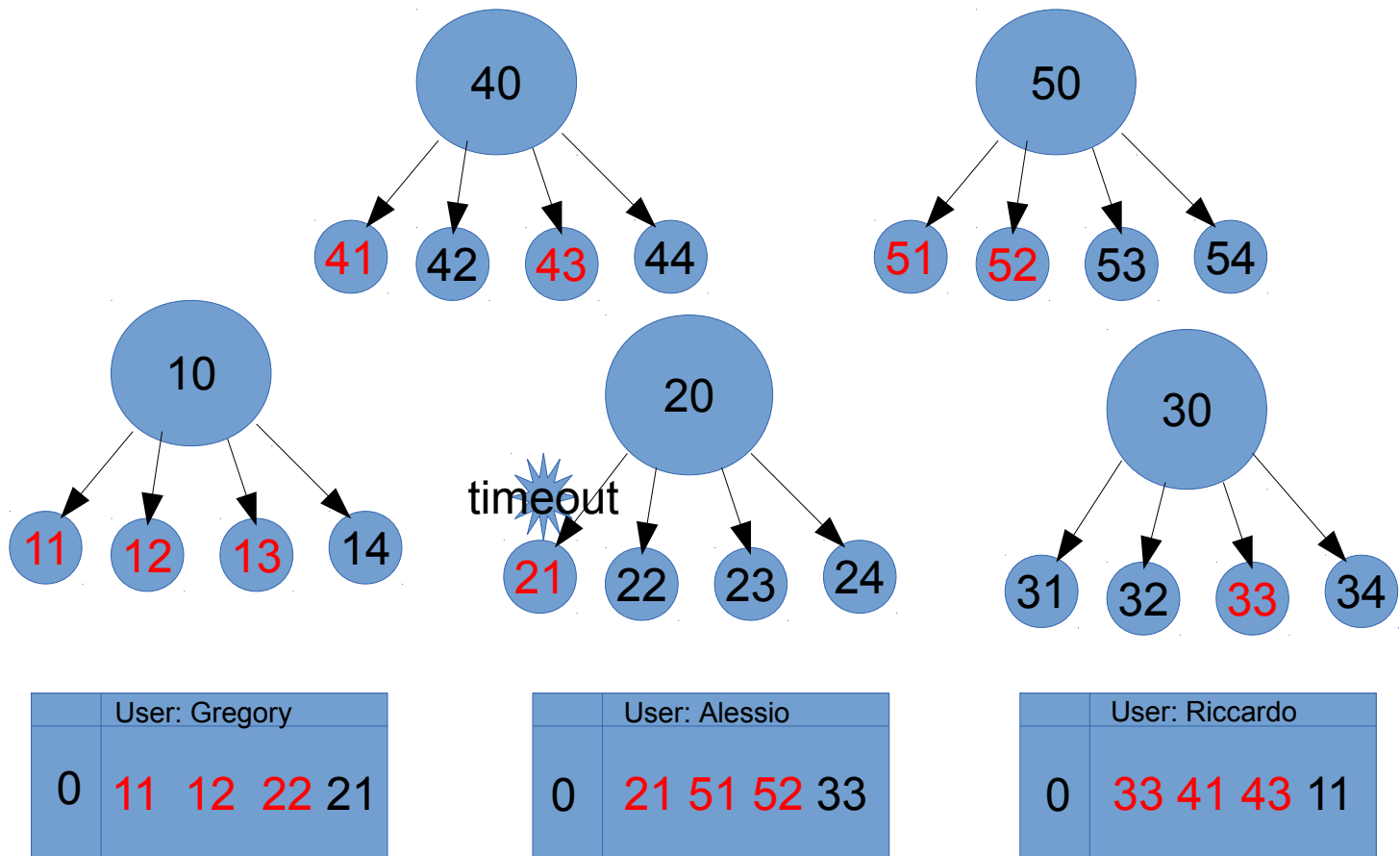
Info → Gli uffici in rosso sono quelli già occupati dalla segnalazione.

Situazione di Deadlock:

- Gregory non può iniziare la sua segnalazione con ID 0 perché 10 non riesce ad ottenere il token dell'ufficio 21 da parte del CallCenter 20 che lo ha già impegnato per la segnalazione 0 di Alessio;
- Alessio non può iniziare la sua segnalazione con ID 0 perché 20 non riesce ad ottenere il token dell'ufficio 33 da parte del CallCenter 30 che lo ha già impegnato per la segnalazione 0 di Riccardo;
- Riccardo non può iniziare la sua segnalazione con ID 0 perché 30 non riesce ad ottenere il token dell'ufficio 11 da parte del CallCenter 10 che lo ha già impegnato per la segnalazione 0 di Gregory;

Situazione di Deadlock:





Parametri globali della DEMO:

- rmregistryAddress: "localhost"
- rmregistryPort: "19000"
- timeout: 20000 *//tempo di attesa (in millisecondi) prima di far scattare il timeout per la
//procedura di rilevazione del deadlock*
- tmin_procedure= 200 *//tempo minimo (in millisecondi) per il disbrigo della pratica da
//parte dell'ufficio*
- tmax_procedure= 500 *//tempo minimo per il disbrigo della pratica da parte dell'ufficio*
- guard_time = 3000 *//tempo di guardia tra un acquisizione di un token e un'altra (inserito
//per poter far capitare sistematicamente una situazione di deadlock)*

Algoritmo di rilevazione del Deadlock (Chandy-Misra-Haas):

→ ordine di partenza delle segnalazioni nel tempo:

- user: Gregory, idSegnalazione=0 , istante t=0; (in secondi)
- user: Alessio, idSegnalazione=0 , istante t=3;
- user: Riccardo, idSegnalazione=0 , istante t=6;

- Con l'ordine di partenza delle segnalazioni indicato, dopo alcuni secondi si arriva alla situazione mostrata in figura che come mostrato alla pagina precedente causa un deadlock.

Successivamente:

- Scatta il timeout al Call Center 20 sull'ufficio 21.
- Call Center 20 verifica che non può liberare 21 perchè la segnalazione che lo blocca attende il token dell'ufficio 33. **Aggiunge quindi 21,33 alla listaUfficiBloccati.**
- Call Center 20 contatta il proprietario dell'ufficio 33 (ovvero 30) e si fa tornare il CallCenter a cui 30 ha dato il token di 33 (in questo caso 30 stesso) e l'id della segnalazione che blocca 33 (in questo caso la segnalazione 0 di Riccardo)
- Call Center 20 invia una probe al CallCenter che ha il token di 33 (ovvero 30) mandando:

- [21,33] (listaUfficiBloccati)
- 20 (id callCenter mittente)
- 0 (segnalazione che blocca l'ufficio 33 sul callCenter destinatario, legata all'utente Riccardo). **Questo suppone che ogni CallCenter conosce gli id delle segnalazioni che bloccano i suoi uffici (anche se queste segnalazioni non appartengono a lui).**
- 10 (id callCenter scatenante, perché nel nostro caso il timeout sull'ufficio 21 al CallCenter 20 è scattato per conto di 20)
- 0 (id segnalazione che ha scatenato il timeout legato al CallCenter del parametro precedente, in questo caso Gregory)

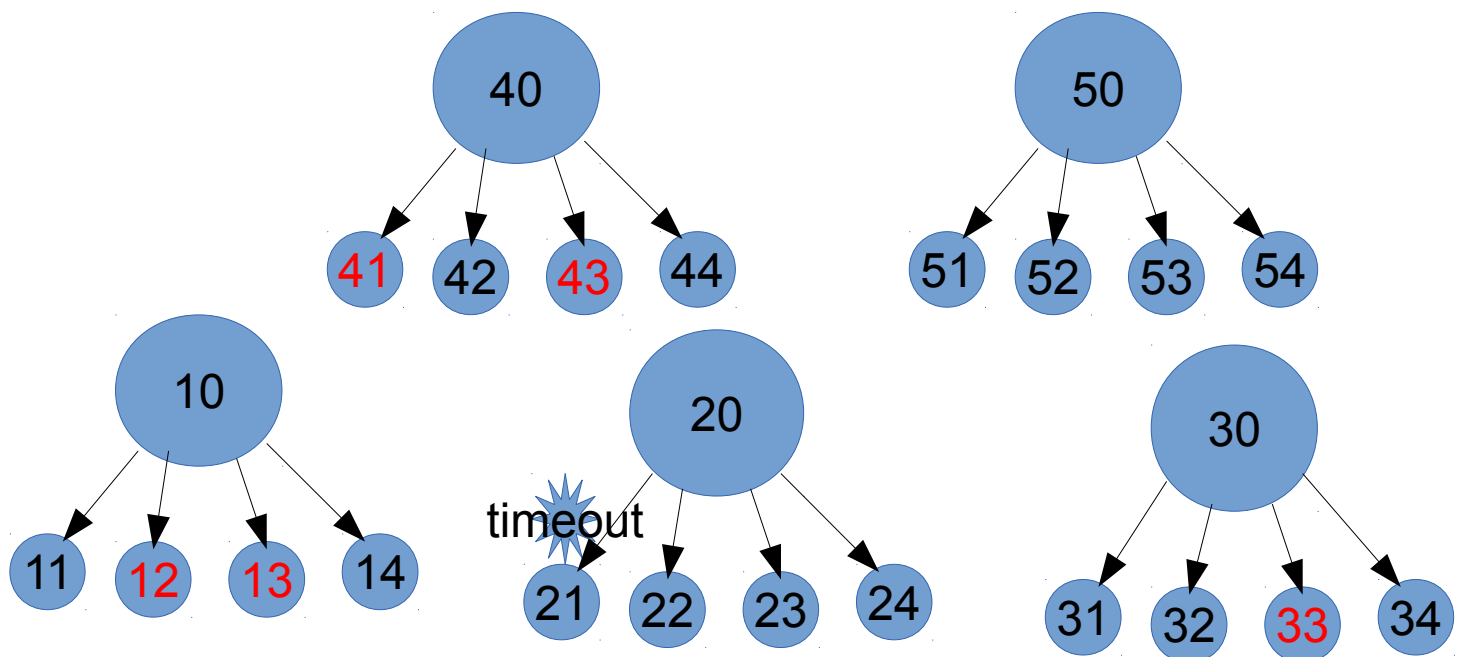
- CallCenter 30 riceve la probe, prende l'ultimo elemento (33) e verifica che non può liberare 33 perchè la segnalazione che lo blocca attende il token dell'ufficio 11. **Aggiunge quindi 11 alla listaUfficiBloccati, controllando prima se 11 è già presente.**
- Call Center 30 contatta il proprietario dell'ufficio 11 (ovvero 10) e si fa tornare il CallCenter a cui 10 ha dato il token di 11 (in questo caso 10 stesso) e l'id della segnalazione che blocca 11 (in questo caso la segnalazione 0 di Gregory)
- Call Center 30 invia una probe al CallCenter che ha il token di 11 (ovvero 10) mandando:

- [21,33, 11] (listaUfficiBloccati)
- 30 (id callCenter mittente)
- 0 (segnalazione che blocca l'ufficio 11 sul callCenter destinatario, legata all'utente Gregory).
- 10 (id callCenter scatenante, perché nel nostro caso il timeout sull'ufficio 21 al CallCenter 20 è scattato per conto di 20)
- 0 (id segnalazione che ha scatenato il timeout legato al CallCenter del parametro precedente, in questo caso Gregory)

- CallCenter 10 riceve la probe, prende l'ultimo elemento (11) e verifica che non può liberare 11 perchè la segnalazione che lo blocca attende il token dell'ufficio 21. **Aggiunge quindi 21 alla listaUfficiBloccati, controllando prima se 21 è già presente. Poichè 21 è già presente nella lista allora significa che c'è stato un DEADLOCK.**

Quindi L'ALGORITMO PRIMA DI OGNI AGGIUNTA ALLA LISTAUFFICIBLOCCATI VERIFICA SE L'ID DELL'UFFICIO CHE STA INSERENDO È GIÀ PRESENTE IN LISTA. Se è già presente allora significa che c'è una situazione di deadlock.

Algoritmo di rimozione dello stallo:



User: Gregory	
0	11 12 22 21

Vengono liberate le risorse:
11, 12, 22
ed eliminata la segnalazione.
Viene rieseguita immediatamente
la segnalazione
0 dell'utente Gregory.

User: Alessio	
0	21 51 52 33

Vengono liberate le risorse:
21, 51, 52
ed eliminata la segnalazione.
Viene notificato all'utente Alessio di
provare a rinviare la segnalazione id 0 più tardi
(non prima comunque di un tempo di guardia)

User: Gregory	
0	11 12 22 21