

**Tomasz Strzeszewski 145 361**  
**Dawid Królak 145 383**

W winiarzy  
S studentów  
B bezpiecznych miejsc

WINIARZ

0. Lamport = id procesu

1. Produkuje wino.

2. Wybierz miejsce

- Wyślij REQ (Lamport+1) do wszystkich innych winiarzy
- Czekaj na W-B odpowiedzi ACK ( Lamport = min(Lamport, ACK.lamport)+1 )
- Jeśli otrzymano REQ, porównaj zegary lamporta i wyślij ACK jeśli ACK.lamport < Lamport

3. Zajmij miejsce i wyślij studentom info o ilości wyprodukowanego wina

4. Zajmuj miejsce B dopóki wyprodukowane wino > 0 (z każdą transakcją wyślij zaktualizowaną ilość wina do studentów)

5. Zwolnij miejsce i wyślij ACK do wszystkich winiarzy ubiegających się o miejsce i powrót do pkt 1.

STUDENT

0. Lamport = id procesu

1. Planuj imprezę na x wina

2. Wybierz studenta

- Po otrzymaniu wiadomości od winiarza dodaj go do lokalnej struktury decyzyjnej z kolejkami
- Powiadamiaj innych studentów o swojej kolejce ( Lamport+1 )
- Po otrzymaniu wiadomości z kolejką od studenta dodaj jego kolejkę do struktury decyzyjnej ( Lamport = min(Lamport, REQ.lamport)+1 )

- Wyślij REQ do wszystkich innych studentów
- Czekaj na S - 1 wiadomości od studentów z większym zegarem Lamporta

4. Wybierz winiarza na podstawie struktury decyzyjnej

- Jeżeli jest pierwszy w kolejce wybierz pierwszego winiarza ze swojej kolejki i powiadom studentów o decyzji
- Jeżeli nie jest pierwszy wybierz winiarza z pozostałych którzy zostaną po eliminacji przez studentów, którzy są wcześniej w kolejce
- Jeżeli nie jest w stanie podjąć decyzji to czekaj na kolejne broadcasty od studentów

5. Transakcja z winiarzem

- Jeśli winiarz oddał mniej wina niż student chciał odebrać to wróć do pkt 1. z x - ilość odebranego wina
- Jeśli winiarz oddał tyle wina ile student chciał przejdź do pkt 6.

6. Student idzie się nawalić i odczekuje losową ilość czasu po czym wraca do pkt 1.