

Zad 12, Mutual Exclusion

Weryfikacja przejść z dowolnej konfiguracji do legalnej wymaga sprawdzenia każdej konfiguracji początkowej. Kolejnym problemem okazało się wygenerowanie *ścieżek* reprezentujących konkretne *wykonanie* (kolejność), które składane jest na podstawie wygenerowanych *rund*.

Szczególną przeszkodą okazało się zarządzanie pamięcią oraz limity z tym związane. Warto zaznaczyć, że wszystkich stanów początkowych jest $(n + 1)^n$. Dodatkowo, każdy taki stan można przekształcać w rundzie na $n!$ sposobów, czyli tyle ile mamy sposobów na zakolejkowanie n procesów. Sposób działania przypomina przeszukiwanie drzewa wszerz.

Pomimo dołożenia wszelkich starań optymalizacyjnych, udało się jedynie znaleźć rozwiązanie dla $n = 4$, gdzie najdłuższa ścieżka była długości 15. W przypadku $n = 5$, zużycie pamięci szybko przekroczyło 8GiB pamięci RAM i uniemożliwiło dalsze obliczenia.

Usprawnienia optymalizacyjne polegały na zaznaczaniu już odwiedzonych wierzchołków oraz zapisywania w nich najdłuższej już znalezionej ścieżki do stanu stabilnego. Oczywistym krokiem było wykluczenie rekursji gdyż szybko doprowadziłaby ona do przepełnienia stosu.