

## Opis rozwiązania

W swoim rozwiązaniu wykorzystuję współbieżność do liczenia algorytmu Brandesa zauważając, że liczenie delty dla różnych źródeł jest kompletnie niezależne. Każdy wątek otrzyma swój zbiór wierzchołków, które ma przerobić.

Rozdzielam dokładnie tę pętlę, która jest w opisie algorytmu ze strony zaznaczona jako „in parallel”. Można ją podzielić równo między wszystkie wątki (każdy dostałby kolejne  $\frac{n}{d}$  wierzchołków), jednakże może się to okazać wysoce nieefektywne, gdyż długość kroku pętli dla danego wierzchołka jako źródła może przejść pesymistycznie wszystkie krawędzie w grafie albo odwiedzić tylko źródło (dla wierzchołka o stopniu wychodzącym 0). Dlatego każdy wątek, gdy nie ma nic do zrobienia „prosi” o nowy wierzchołek do przerobienia. Współbieżne „proszenie” o kolejny wierzchołek kontroluje wzajemne wykluczanie sekcji krytycznej realizowane za pomocą mutexu.

## Wydajność

Testy były przeprowadzane na moim komputerze oraz na students.

Mój komputer (Intel i5, 4 rdzenie):			Students:		
Wątki	Czas	Przyspieszenie	Wątki	Czas	Przyspieszenie
1	2.207	-	1	7.274	-
2	1.301	1.6963	2	4.098	1.77496
3	1.103	2.0009	3	2.411	3.01701
4	1.010	2.1851	4	1.990	3.6552
5	1.030	2.1426	5	1.733	4.1973
6	1.032	2.1385	6	1.593	4.5662
7	1.048	2.1058	7	1.359	5.3524
8	1.083	2.0378	8	1.261	5.7684