

## Implementacja:

Moja implementacja wykorzystuje  $2 * N$  mutexow, po jednym dla adoratorow kazdego wierzcholka, po jednym dla zmiany ilosci poszukiwanych par oraz dodatkowy mutex dla kolejki z ktorej watki pobieraja wierzcholek do przerobienia.

Kazdy watek pobiera wierzcholek z kolejki lockujac sie na mutexie kolejki, nastepnie dla tego wierzcholka szuka b wierzcholkow ktore bedzie adorowal, b jest na poczatku wartoscia funkcji bvalue dla tego wierzcholka ale po jednym przerobieniu wierzcholka staje sie liczba razy ktora ten wierzcholek mial "anulowana" adoracje innego wierzcholka, na zmiennej tej podczas modyfikacji lockowany jest mutex. Przeglada on nastepnie swoich sasiadow. W ramach optymalizacji sortuje partial\_sortem czesc krawedzi wychodzaca z wierzcholka, przez co kazda krawedz bedzie rozpatrzona tylko raz, gdy dojde do konca posortowanej czesci, sortuje dalszy kawalek wektora itd (czescia sortowana wynosi  $bvalue(v) * 7$ ). Za kazdym razem gdy przegladam sasiada lockuje sie na jego mutexie by nikt go w tym czasie nie zmodyfikowal. Jezeli krawedz do tego wierzcholka jest lepsza od najgorszej adorujacej jej krawedzi to ustawiam ze go adoruje I jezeli anulowalem w ten sposob adoracje innego wierzcholka to wrzucam go do kolejki, oddzielnej dla kazdego watku, po czym zwalniam wczesniej zajety mutex. Po zakonczeniu wykonania wszystkich watkow lacze wszystkie ich kolejki do ktorych wrzucali wierzcholki z anulowana adoracja w jedna kolejke z unikatowymi wierzcholkami ktora bedzie przerabiana w kolejnej iteracji.

## Czasy I Przyspieszenia (bez wczytywania, mierzone przez chrono::high resolution clock) :

### as-Skitter:

- P\_1: 20.8973s      S\_1: 1
- P\_2: 16.4783s      S\_2: 1,2681
- P\_3: 16.2561s      S\_3: 1,2855
- P\_4: 15.2460s      S\_4: 1,3706
- P\_5: 16.5014s      S\_5: 1,2663
- P\_6: 15.1302s      S\_6: 1,3811
- P\_7: 12.1417s      S\_7: 1,7211
- P\_8: 14.1275s      S\_8: 1,4791

### RoadNet-PA:

- P\_1: 17.6620s      S\_1: 1
- P\_2: 15.7784s      S\_2: 1,1193
- P\_3: 14.2247s      S\_3: 1,2416
- P\_4: 15.8270s      S\_4: 1,1159
- P\_5: 14.6635s      S\_5: 1,2044
- P\_6: 13.9107s      S\_6: 1,2696
- P\_7: 12.4702s      S\_7: 1,4163
- P\_8: 13.5467s      S\_8: 1,3037