



**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

**Drzewa wszystkich najkrótszych ścieżek - algorytm
Dijkstry
Aplikacja PGAS**

Arkadiusz Kasprzak, Aleksandra Poręba



- 1 Algorytm Dijkstry - wprowadzenie**
- 2 Budowa i działanie projektu**
- 3 Kompilacja i uruchomienie**



AGH Algorytm Dijkstry - wprowadzenie

- Cel: znalezienie najkrótszych ścieżek z wybranego wierzchołka grafu do wszystkich pozostałych wierzchołków.
- Operujemy na grafie skierowanym lub nieskierowanym o nieujemnych wagach krawędzi.
- Algorytm zachłanny - w każdym kroku algorytmu wybierany jest wierzchołek o najmniejszej wartości kosztu.
- Możliwość znalezienia zarówno najkrótszych ścieżek, jak i ich kosztów.



Program podzielony został na części:

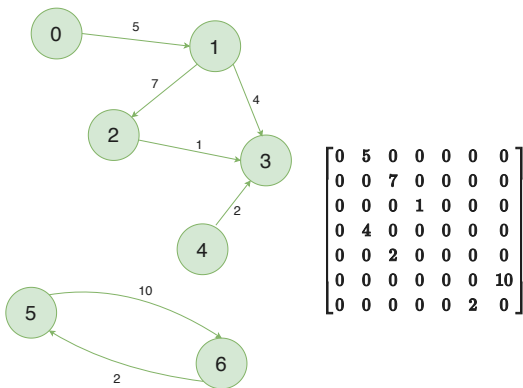
- wczytanie danych wejściowych,
- zapis kolumn do sąsiedztwa odpowiednich procesów,
- obliczanie odległości i ścieżek - algorytm Dijkstry,
- zapisy wyników do pliku oraz zakończenie.

Szczegółowy opis wraz ze schematami blokowymi zawarty został w dokumentacji projektu.



AGH Działanie projektu - dane wejściowe

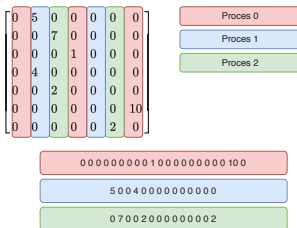
- Dane podawane w postaci tzw. macierzy sąsiedztwa.
- Są one wczytywane z pliku przez proces o numerze 0.





AGH Działanie projektu - podział danych wejściowych

- Dane dzielone są kolumnami równomiernie pomiędzy wszystkie procesy.
- Jeśli podział równomierny jest niemożliwy, pozostałe k kolumn dzielone jest między pierwsze k procesów.
- Procesy otrzymują kolumny o indeksach
$$\text{indeks_kolumny} \bmod \text{liczba_procesów} == \text{numer_procesu}$$



- Operujemy na trzech tablicach: tablicy kosztów, tablicy poprzedników, oraz tablicy informującej, czy wierzchołek został już przetworzony.
- Są one współdzielone pomiędzy procesami, ale każdy z nich będzie wypełniał pozycje tylko dla wierzchołków w jego sąsiedztwie.
- Tablica kosztów inicjalizowana nieskończonościami, tablica poprzedników - wartościami -1, a tablica przetworzonych zerami.
- Koszt dla wierzchołka źródłowego ustawiany na 0.

Tablica kosztów							
0	∞	∞	∞	∞	∞	∞	∞

Tablica poprzedników							
-1	-1	-1	-1	-1	-1	-1	-1

Tablica przetworzonych							
0	0	0	0	0	0	0	0

Figure: Przykładowy początkowy stan tablic

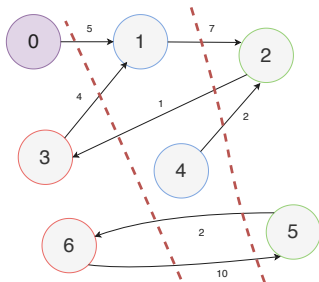


AGH Działanie projektu - przebieg algorytmu

W dalszej części programu powtarzamy poniższe czynności aż wszystkie wierzchołki zostaną przetworzone:

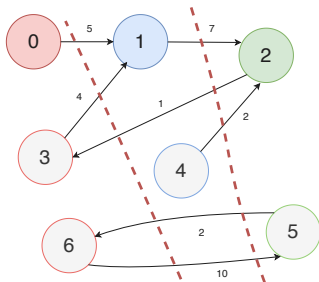
- każdy proces spośród przydzielonych mu nieprzetworzonych jeszcze wierzchołków wybiera ten o najmniejszej wartości kosztu
- spośród wszystkich odległości wybierana jest ta najmniejsza (operacja `upc_all_reduce`). Odnajdowany zostaje indeks wierzchołka i zostaje on zaznaczony jako przetworzony
- na podstawie wylosowanego wierzchołka przeprowadzana jest aktualizacja w tabelach kosztów i poprzedników

Po zakończeniu działania algorytmu wyniki algorytmu zapisywane są przez proces 0 do pliku.

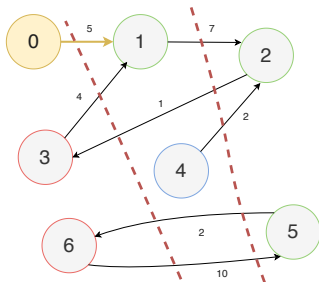


0	∞	∞	∞	∞	∞	∞
---	----------	----------	----------	----------	----------	----------

Sytuacja początkowa. Na fioletowo został zaznaczony wierzchołek źródłowy. Czerwoną linią pokazany został podział grafu między procesy. Na dole znajduje się tablica kosztów.



Wybór nieprzetworzonych wierzchołków o najniższym koszcie dotarcia (w tablicy) w każdym z procesów.



Wybór wierzchołka o globalnie najniższym koszcie (`upc_all_reduce`) i zaznaczenie go jako przetworzony. Aktualizacja kosztów i poprzedników.



- Podział grafu pomiędzy poszczególne procesy jest przedstawiany użytkownikowi.

```
Proces 0 posiada 4 kolumn o indeksach: 0, 3, 6, 9,  
Proces 1 posiada 3 kolumn o indeksach: 1, 4, 7,  
Proces 2 posiada 3 kolumn o indeksach: 2, 5, 8,
```



- Przejście do katalogu `build_uni`
- Przygotowanie środowiska pracowni za pomocą skryptu `setup.sh`
- Wykonanie polecenia `make`
- Wykonanie polecenia `make run` z opcjonalnymi argumentami określającymi: plik z węzłami, ilość procesów, numer wierzchołka źródłowego oraz ścieżkę do pliku z danymi
- Wynik dostępny w pliku `resultsUPC.txt`
- Proces szczegółowo opisany w dokumentacji