Algorytmy przeszukiwania grafów. Konstrukcja drzew rozpinających.Metoda BFS

Izolda Gorgol

wyciąg z prezentacji (wykład V, 10.06.2007 r.)

Algorytm BFS

WEJŚCIE: graf G; wierzchołek v, od którego zaczynamy przeszukiwanie
WYJŚCIE: LISTA wierzchołków w kolejności odwiedzania
Zmienna pomocnicza KOLEJKA
$BFS(G, v)$

1. $KOLEJKA := \emptyset$;
2. $LISTA := \emptyset$;
3. dołącz v do KOLEJKI;
4. **dopóki** KOLEJKA$\neq \emptyset$ **wykonuj**
   a) usuń p z KOLEJKI; odwiedź p; dołącz p do LISTY;
   b) **dla wszystkich** sąsiadów u wierzchołka p **wykonaj**
      i. **jeśli** u nie był jeszcze wykorzystany **to** dołącz u do KOLEJKI.

Konstrukcja drzewa rozpinającego przy użyciu algorytmu BFS

WEJŚCIE: spójny graf G; wierzchołek v - korzeń drzewa
WYJŚCIE: $E(T)$ - zbiór krawędzi drzewa T
Zmienna pomocnicza KOLEJKA
$TBFS(G, v)$

1. $KOLEJKA := \emptyset$;
2. $E(T) := \emptyset$;
3. dołącz v do KOLEJKI;
4. **dopóki** KOLEJKA$\neq \emptyset$ **wykonuj**
   a) usuń p z KOLEJKI; odwiedź p;
   b) **dla wszystkich** sąsiadów u wierzchołka p **wykonaj**
      i. **jeśli** u nie był jeszcze wykorzystany **to** dołącz u do KOLEJKI i dołącz up do $E(T)$.

Porównanie algorytmów DSF i BFS

1. złożoność obliczeniowa obydwu jest rzędu $O(|V(G)| + |E(G)|)$
2. algorytm DSF, jako rekurencyjny, jest prostszy w zapisie (i implementacji)
3. algorytm DSF w prosty sposób konstruuje ścieżki pomiędzy wierzchołkami
4. algorytm BSF konstruuje najkrótsze ścieżki pomiędzy wierzchołkami (czyli może być użyty do znalezienia odległości pomiędzy wierzchołkami)