Algorytmy i struktury danych

Karol Bonenberg

Algorytmy z powracaniem

10.04.2005

I. Wstęp

W poniższym ćwiczeniu przestawiono zagadnienia dotyczące idei działania algorytmów z powracaniem oraz problemów znajdowania cyklu Eulera i Hamiltona w grafie nieskierowanym oraz ich przynależności do odpowiednich klas złożoności obliczeniowej.

- **Idea działania algorytmów z powracaniem**

Idea działania algorytmów z powracaniem polega na przechodzeniu przez jakieś połączone elementy w celu osiągnięcia jakiejś konkretnej kolejności odwiedzin. W przypadku, gdy przechodzenie dalej już nie jest możliwe, algorytm ‘cofa’ się do poprzedniego elementu, i z niego wybiera inną ‘drogę’. Dzieje się tak, aż do momentu, gdy znajdziemy założony ciąg przejść lub wyczerpie się liczba kombinacji. Z reguły algorytmy z powracaniem mają charakter rekurencyjny.

- **Problem znajdowania cyklu Eulera i Hamiltona w grafie nieskierowanym oraz ich przynależności do odpowiednich klas złożoności obliczeniowej**

Aby w grafie istniał cykl Eulera, graf musi być spójny i maksymalnie dwa wierzchołki mogą być stopnia nieparzystego. Aby znaleźć taki cykl możemy posłużyć się następującym algorytmem. Korzystamy w nim z dwóch stosów: jednego głównego pamiętającego wierzchołki, które być może będą elementami cyklu Eulera, drugiego do pamiętania kolejnych wierzchołków cyklu.

Główną część algorytmu stanowi pętla wykonywana dopóki istniej jeszcze wierzchołki na głównym stosie, tzn. istnieją kandydaci na element cyklu. Dla danego wybranego wierzchołka sprawdzane s wierzchołki z nim incydentne i jeśli takie istniej dana krawędź jest usuwana z grafu. Algorytm ten ma złożoność O(k), jest, więc wielomianowy w przeciwieństwie do algorytmu znajdowania cyklu Hamiltona w grafie, który należy do klasy problemów NP -zupełnych, jego złożoność nie jest, więc wielomianowa.

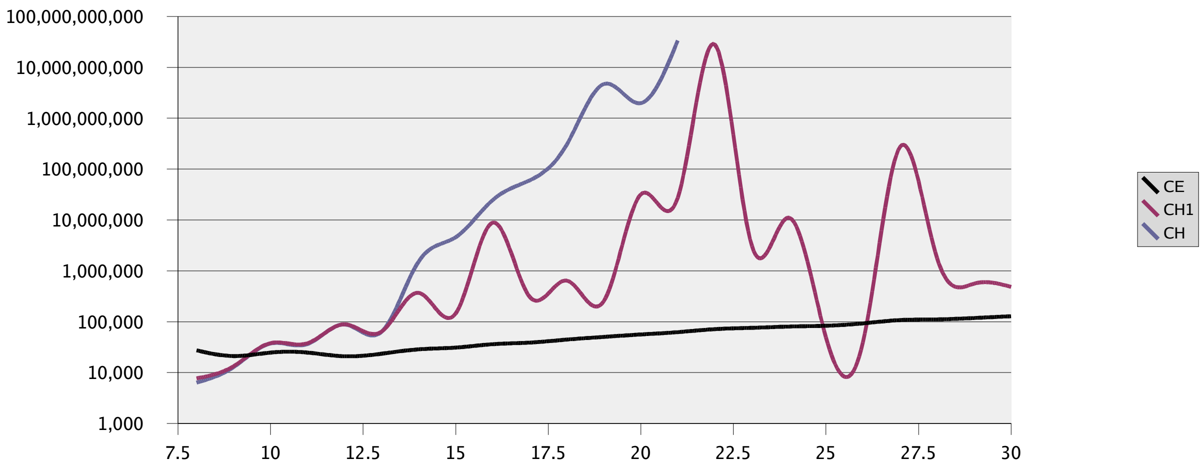
- **Idea algorytmu z powracaniem dla problemu znajdowania cyklu Hamiltona.**

Algorytm znajdowania cyklu Hamiltona ma postać rekurencyjną. Korzystamy tutaj z dwóch dodatkowych tablic: jednej przechowującej aktualna 'ścieżkę' w drzewie wyszukiwań w grafie, drugą przechowującą wartości logiczne odpowiadające każdemu wierzchołkowi w ścieżce - mówi ona czy dany wierzchołek może być użyty jako element cyklu. Funkcja rekurencyjna jest wywoływana z parametrem oznaczającym indeks kolejnego elementu w ścieżce, (która później być może stworzy cykl). W funkcji jest pętla, która sprawdza każdą możliwą kombinację wierzchołków incydentnych do danego (przekazanego jako parametr). Jeżeli okaże się, że jest to ostatni wierzchołek, a ponadto jest równy pierwszemu w ścieżce, to został utworzony cykl Hamiltona. Jeśli nie, sprawdzany jest kolejny wierzchołek (wybrany w głównym warunku pętli) i funkcja jest wywoływana z tym wierzchołkiem jako parametrem.

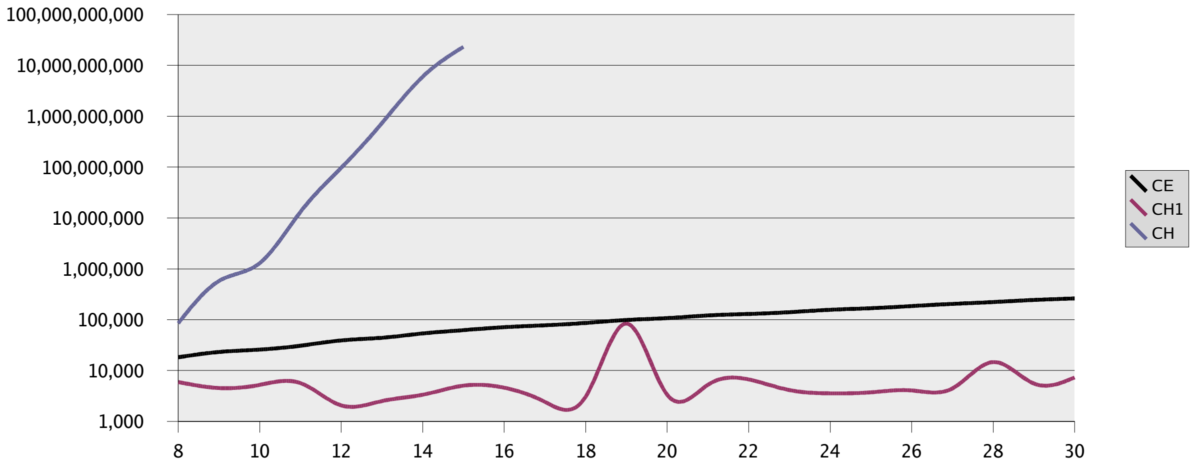
Mechanizm powracania istnieje w powyższym algorytmie dzięki rekurencji, która sama w sobie ma ten mechanizm. Tzn., gdy skończy się wykonywanie pętli dla k, wykonywana jest pętla (albo raczej kontynuowane jej wykonanie) dla k-1, ponieważ ta wartość była pamiętana na stosie programu. Ma to taki sam efekt jakbyśmy stworzyli ten algorytm iteracyjnie nie używając rekurencji, wtedy indeks k musiałby być zmniejszany (lub zwiększany w zależności od rezultatu), co byłoby częścią mechanizmu powracania.

II.Wyniki pomiarów

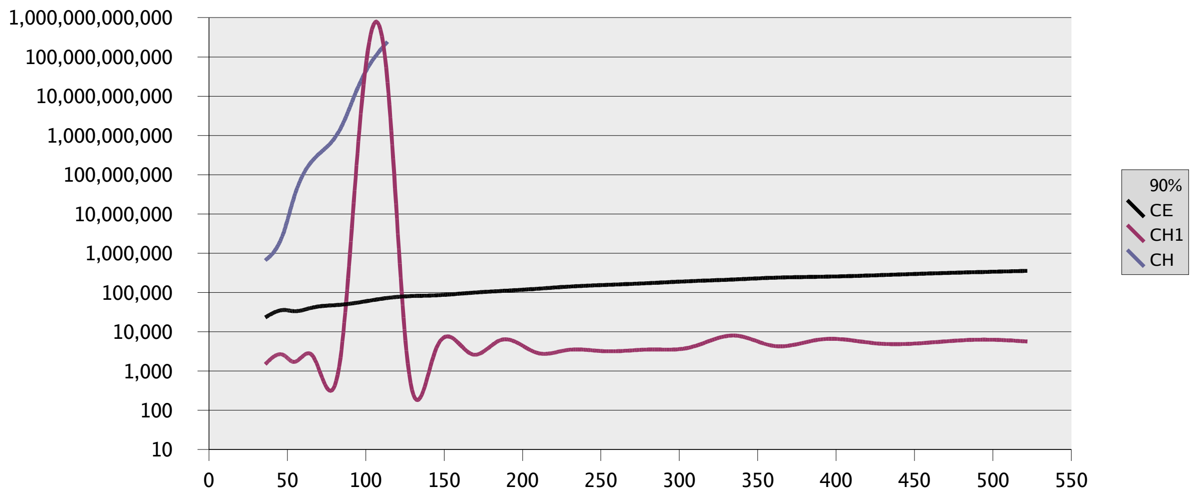
tCE= f(n), tCH1= f(n), tCH= f(n) dla w=30%

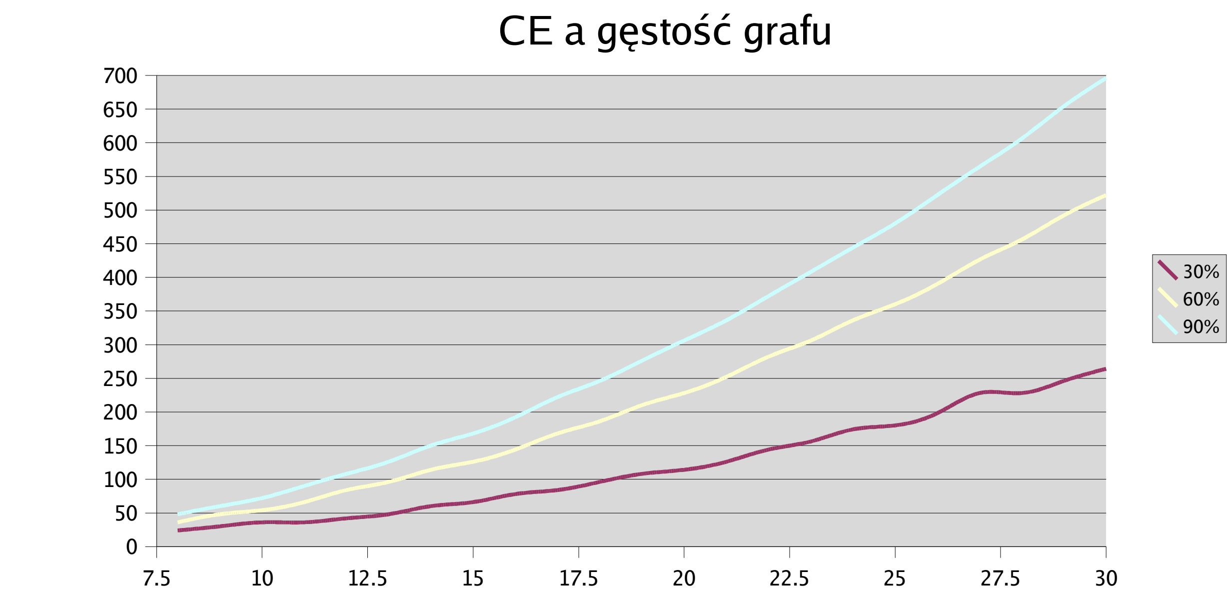


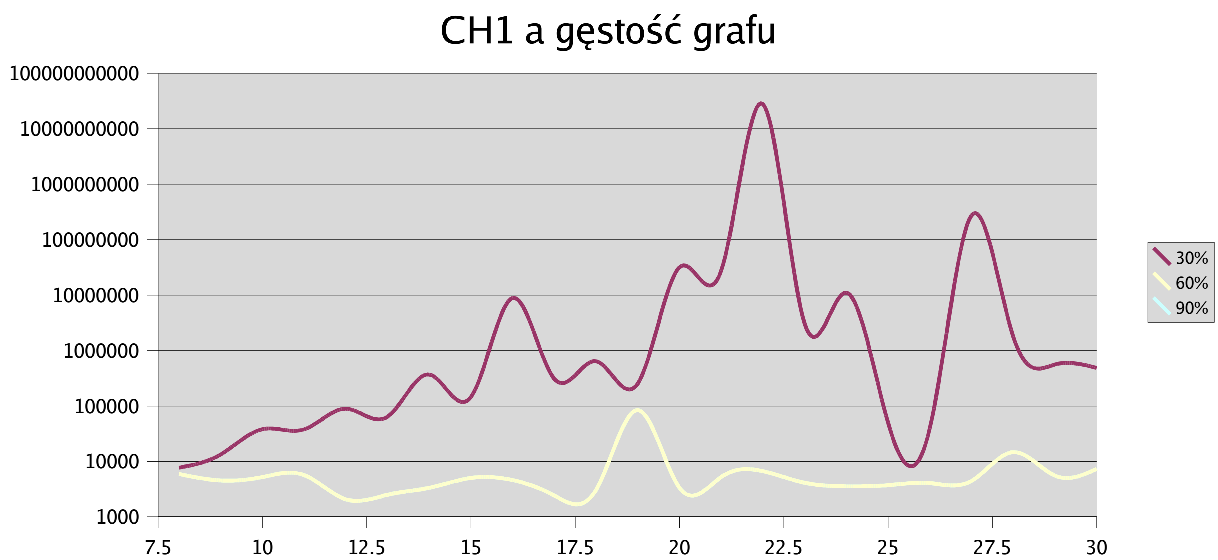
tCE= f(n), tCH1= f(n), tCH= f(n) dla w=60%

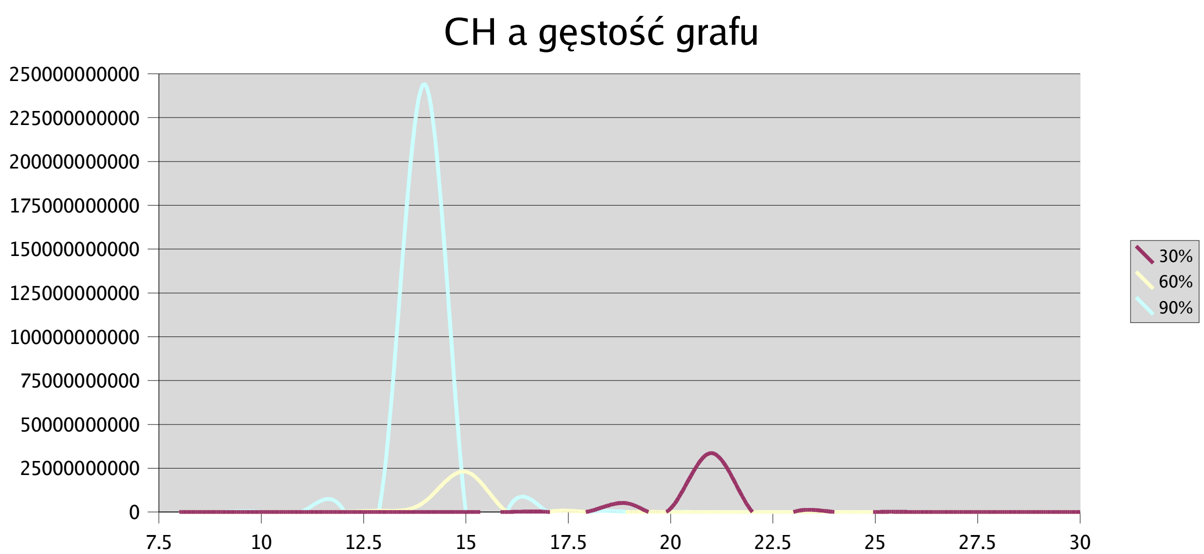


tCE= f(n), tCH1= f(n), tCH= f(n) dla w=90%




CE a gęstość grafu
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
7.5
10
12.5
15
17.5
20
22.5
25
27.5
30
30%
60%
90%


CH1 a gęstość grafu
100000000000
10000000000
1000000000
100000000
10000000
1000000
100000
10000
1000
7.5
10
12.5
15
17.5
20
22.5
25
27.5
30
30%
60%
90%


CH a gęstość grafu
250000000000
225000000000
200000000000
175000000000
150000000000
125000000000
100000000000
75000000000
50000000000
25000000000
0
7.5
10
12.5
15
17.5
20
22.5
25
27.5
30
30%
60%
90%

III.Odpowiedzi

1. Zadanie

- **Do klasy jakich problemów należy problem znajdowania cyklu Eulera?**

Złożoność jest równa $O(n^k)$.Problem należy do klasy problemów łatwych – istnieje algorytm odnajdujący cykl Eulera w czasie wielomianiowym.

- **Dlaczego w testowanych grafach istniał cykl Eulera?**
  Cykle Eulera zawarte były w każdym z wygenerowanych przez nas grafów – algorytm generujący dodaje za każdym razem 3 nowe wierzchołki i łączy „każdy z każdym", wprowadzając za każdym razem parzystą liczbę krawędzi.

- **Warunek konieczny i dostateczny istnienia cyklu Eulera?**

Warunkiem koniecznym jest spójność grafu, nie można bowiem odwiedzić wszystkich dróg jeżeli do niektórych nie ma „połączenia".

Warunekdostateczny dla grafu nieskierowanego to parzysty stopień każdego z wierzchołków $\sum_{v \in V} d(v) = 2m$ dla grafu skierowanego – równa liczba krawędzi wchodzących i wychodzących z danego wierzchołka in(v) = out(v)

- **Określić i uzasadnić złożoność obliczeniową algorytmu (CE)?**

Wersja algorytmu użyta do uzyskania wyników umieszcza wierzchołek na stosie i usuwa krawędź z grafu lub, w zależności od sytuacji, przenosi wierzchołek z jednego stosu na drugi.

Wynika z tego, że liczba iteracji algorytmu jest rzędu O(n). Używając listy incydencji mamy możliwość usuwania poszczególnych krawędzi w czasie ograniczonym pewną stałą. Uzyskujemy dzięki temu iteracje wykonywane w stałym czasie, co pozostawia złożoność algorytmu na poziomie O(n)

- **Do jakiej klasy należy problem odnajdywania cyklu Hamiltona?**

Problem ten należy do klasy problemów NP.-zupełnych, tzn. takich ,dla których nie są znane (najprawdopodobniej nie istnieją) algorytmy rozwiązujące je w czasie wielomianowym. Istnieje takiego algorytmu dla choćby jednego problemu z tej klasy, implikowałoby istnieje takich algorytmów dla wszystkich problemów. Należy pamiętać, iż zakładamy różność klas P i NP.

- **Określić i uzasadnić złożoność algorytmu znajdowania HA i HA1**

Ponieważ możliwa liczba permutacji n wierzchołków wynosi n!, dlatego też, w przypadku pesymistycznym, gdy szukany cykl Hamiltona znajduje się w ostatniej z nich, algorytm musi sprawdzić n! możliwych rozwiązań, [zarówno, gdy szukamy jednego jak i więcej cykli].

Złożoność jest zatem rzędu $\Omega(n!)$. W przypadku optymistycznym, gdy szukamy 1 cyklu Hamiltona i odnajdujemy go w pierwszej permutacji, algorytm przegląda n wierzchołków co daje złożoność O (n). Gdy jednak szukamy wielu cykli, i tak trzeba sprawdzić wszystkie permutacje.

2. Zadanie

- **Jaki jest wpływ liczby wierzchołków, a jaki liczby krawędzi na czas działania metody?**

W przypadku optymistycznym, gdzie znalezienie kolejnego wierzchołka odbywa się w takim samym czasie, czas działania tego algorytmu zależy tylko i wyłącznie od ilości krawędzi. Oznacza to, że dla wzrostu ilości krawędzi rośnie czas poszukiwania cyklu Eulera, a dla zmniejszania ilości krawędzi czas maleje.

Jeżeli weźmiemy jednak pod uwage reprezentację grafu i czas wyszukiwania następnego wierzchołka to ilość wierzchołków ma znaczenie.

- **Czy reprezentacja grafu wpływa na złożoność metody?**

Jeżeli bazujmy na grafie skierowanym to istotne dla nas są tylko następniki danego wierzchołka. W tym przypadku doskonale sprawdza się lista następników. Znalezienie kolejnego wierzchołka następuje w stałym czasie bo jest on zawsze pierwszy na liście następników. Usunięcie go przesuwa "drugi" następnik w miejsce pierwszego. Złożoność szukania cyklu Eulera przy tej reprezentacji czasu wynosi O(n)

Stosując macierz sąsiedztwa w pesymistycznym przypadku należy przejrzeć cały wiersz O(n). Znalezienie cyklu Eulera wymaga m-krotnego szukania następników więc złożonośc znajdywania tego cyklu wynosi O(m*n).

W przypadku grafu nieskierowanego najsukteczniejszą metodą reprezentacji jest lista następników. Dostęp do kolejnego wierzchołka ma złożoność O(1) czyli jest stały. Macierz sąsiedstwa jest prostsza w implementacji ale czas dostępu do następnego wierzchołka jest dłuższy. Wynika to z tego że dla pesymistycznego przypadku musimy przejrzeć caly wiersz (lub cała kolumne) czyli złożoność tej operacji to O(n). Znalezienie cyklu Eulera ma złożoność O(m*n).

W przypadku stosowania listy łuków znalezienie cyklu Eulera ma złożoność $O(m^2)$.

3. Zadanie

- **Jaki jest wpływ liczby wierzchołków, a jaki liczby krawędzi na czas działania metody?**

Szybkość znajdywania wszystkich cykli Hamiltona zależy od ilości wierzchołków i od ilości krawędzi (czyli gęstości):

- Ilość wierzchołków wpływa wykładniczo na czas znajdywania cyklu Hamiltona. Wynika to z złożoności tej procedury która sprawdza za każdym razem wszystkie możliwe permutacje wierzchołków.

- Ilość krawędzi wpływa na ilość połączeń między wierzchołkami. Zwiększa to ilość możliwych kombinacji wierzchołków połączonych ze sobą.

IV.Podsumowanie

Czy algorytm z powracaniem może być stosowany wyłącznie dla problemu znajdowania cyklu Hamiltona? Jaka jest ogólna idea działania algorytmu z powracaniem?

Algorytmy z powracaniem służą do rozwiązywania problemów NP. zupełnych. Wykorzystując metodę naiwną sprawdzamy wszystkie możliwe permutacje (kombinacje wierzchołków).
Aby zwiększyć wydajność takich algorytmów z reguły ograniczamy liczbę możliwych kombinacji. W powyższym zadaniu ograniczyliśmy się do wierzchołków spełniających pewne wymagania, czyli połączonych krawędzią oraz tworzących określony cykl.