Zadania domowe. Zestaw 3.1

Maciej Poleski

11 grudnia 2012

1

Proces nie zakończy się jeżeli dojdziemy do cyklu. Czyli naszym zadaniem jest stwierdzenie z których wierzchołków nie da się dotrzeć do cyklu. Generalnie zachodzi zależność, że jeżeli z wierzchołka x da się dotrzeć do wierzchołka y z którego da się dotrzeć do cyklu, to z wierzchołka x da się dotrzeć do cyklu. Zastosuję standardowy DFS szukający cyklu + tą przechodniość.

```
C[v] <- czy z wierzchołka v da się dotrzeć do cyklu
state[0..n] <- READY
dfs(v):
   if state[v] = BUSY:
       return CYKL
   else if state[v] = DONE:
       return C[v]
   state[v] <- BUSY
   for each (v,u) ∈ E:
       if dfs(u) = CYKL:
           state[v] <- DONE
           C[v] <- CYKL
           return C[v]
   state[v] <- DONE
   C[v] <- OK
   return C[v]

for each v ∈ V:
   if dfs(v) = OK:
       print v
```

Koszt algorytmu to jednokrotne przejście po całym grafie. $O(n+m)$

2

Wewnątrz każdej silnie spójnej składowej można podróżować bez ograniczeń (z każdego wierzchołka możemy dotrzeć do każdego innego). Skoro tak to problem sprowadza się do odbycia podróży z silnie spójnej składowej wierzchołka x do składowej wierzchołka y (lub odwrotnie). Ale teraz już w DAG-u. Zadanie sprowadza się w takim razie do rozstrzygnięcia czy DAG jest porządkiem liniowym. (Jeżeli jest - oczywiście graf jest średnio-spójny, jeżeli nie - istnieją dwa wierzchołki nieporównywalne - czyli nie jest średnio-spójny).

```
Wyznacz podział na silnie spójne składowe
S[s] <- ∅ - składowa do której istnieje krawędź z składowej s
U[s] <- ∅ - składowa z której istnieje krawędź do składowej s
for each (v,u) ∈ E:
   if v jest w innej składowej niż u:
       if S[id-składowej(v)] != ∅ and S[id-składowej(v)] != id-składowej(u):
           GRAF NIE JEST ŚREDNIO-SPÓJNY
       S[id-składowej(v)] <- id-składowej(u)
       if U[id-składowej(u)] != ∅ and U[id-składowej(u)] != id-składowej(v):
           GRAF NIE JEST ŚREDNIO-SPÓJNY
       U[id-składowej(u)] <- id-składowej(v)
CS <- 0
CU <- 0
for each s będącego id jakiejś składowej:
   if S[s] = ∅:
       CS <- CS + 1
   if U[s] = ∅:
       CU <- CU + 1

if CU != 1 or CS != 1:
   GRAF NIE JEST ŚREDNIO-SPÓJNY
GRAF JEST ŚREDNIO-SPÓJNY
```

Najpierw wyznaczam podział na silnie spójne składowe. Przykładowe rozwiązanie tego problemu można znaleźć w zadaniach Q i R. Następnie sprawdzam czy porządek jest liniowy (czyli czy dla każdej składowej istnieje dokładnie jeden poprzednik i dokładnie jeden następnik z wyjątkiem dokładnie jednej składowej która nie ma poprzednika i dokładnie jednej która nie ma następnika). Podział na SSS $O(n+m)$, pętla po krawędziach $O(m)$, pętla po składowych $O(n)$. Całość $O(n+m)$.

3

4

Przedstawię rozwiązanie w złożoności $O(n^2+m)$. Problem można sprowadzić do 2-SAT-u. Dla każdej krawędzi {u,v} mamy formułę $(u \vee v)$ mówiącą że co najmniej jedno z miast u lub v musi być stolicą. Następnie dla każdej pary miast {u,v} z tego samego województwa mamy formułę $\neg u \vee \neg v$ mówiącą że co najwyżej jedno z miast u lub v może być stolicą. W ten sposób gwarantujemy że w województwie nie istnieje para miast które są stolicą, czyli że istnieje co najwyżej jedna stolica (jeżeli uzyskamy rozwiązanie instancji w którym nie istnieje stolica, to możemy bezpiecznie dodać jedną dowolną - siłą rzeczy nie mamy szans uszkodzić w ten sposób rozwiązania). Mamy w ten sposób $n^2 + m$ formuł i n zmiennych. Możemy znaleźć rozwiązanie tej instancji w czasie $O(n + n^2 + m) = O(n^2 + m)$. Zasadniczo chcemy wiedzieć jedynie czy rozwiązanie istnieje. Możemy wykorzystać zadanie R które rozwiązuje ten problem. (Szkoda przeklejać kodu - to jest dokładnie to samo).