Zadania domowe. Zestaw 1.3

Maciej Poleski

6 listopada 2012

1

Zakładam że rozpoczyna gracz A. Czyli pytanie brzmi: ile jest pozycji wygrywających dla gracza rozpoczynającego grę. `B` jest funkcją zwracającą wartość wyrażenia logicznego podanego w argumencie (`true` lub `false`).

```
V <- zbiór wierzchołków
T[V] <- czy gracz zaczynający z wierzchołkiem v wygrywa
           (zostanie obliczone)
a <- 0 (ile jest pozycji wygrywających (dla pierwszego gracza))
for v in V w odwrotnej kolejności topologicznej:
    T[v] <- B(istnieje krawędź z wierzchołka v do wierzchołka w
                takiego że T[w] = false)
    if T[v] = true:
        a <- a + 1
```

Kolejność odwrotna topologiczna to kolejność w której "obgryzamy" graf. Czyli zaczynamy od wierzchołków które nie mają krawędzi wyjściowych. Jeżeli jesteśmy w stanie pozostawić drugiemu graczowi stan przegrywający, to nasz stan jest wygrywający. (Poprzez wykonanie posunięcia, które spowoduje powstanie stanu przygrywającego i pozostawienie drugiego gracza w tym stanie). Jeżeli nie jesteśmy w stanie pozostawić drugiemu graczowi stanu przegrywającego to znaczy że albo każdy ruch doprowadzi do stanu wygrywającego (i pozostawimy drugiego gracza w stanie wygrywającym), albo nie mamy możliwości wykonania żadnego ruchu (czyli przegrywamy). Odpowiedź to $\frac{a}{n}$ (taka jest szansa, że wybrane zostanie pole wygrywające dla gracza A, czyli przegrywające dla gracza B). Złożoność głównej pętli jest oczywiście równa $|V| = n$ sprawdzenie czy istnieje krawędź kosztuje nas łącznie m. Sortowanie topologiczne (omówione na MP) kosztuje $n + m$. Całość $O(n + m)$.

2

3

```
V <- zbiór wierzchołków
A[V] <- 1 (ilość zbiorów niezależnych w poddrzewie v
            takich że v nie należy do tych zbiorów)
B[V] <- 1 (ilość zbiorów niezależnych w poddrzewie v
            takich że v należy do tych zbiorów)
calc(v):
```

```
for each w in V taki że w jest synem v:
    calc(w)
for each w in V taki że w jest synem v:
    A[v] <- A[v] * (A[w] + B[w])
    B[v] <- B[v] * A[w]
```

Wywołujemy tę funkcję dla dowolnego wybranego wierzchołka v. Odpowiedzią będzie wtedy `A[v] + B[v]`. Obie pętle będą miały łącznie dokładnie $n-1$ obiegów. W efekcie złożoność $O(n)$. Funkcja `calc` jest wykonywana w kolejności postorder. Dla każdego poddrzewa o korzeniu v oblicza ona liczbę zbiorów niezależnych przy założeniu że v do nich należy oraz to samo przy założeniu że nie należy. Jeżeli wierzchołek v należy, to wtedy żadne z jego dzieci nie może należeć. Jeżeli nie należy, to nie ma to znaczenia. Sama ilość tych zbiorów to po prostu ilość wszystkich możliwych kombinacji doboru zbiorów w poddrzewach dzieci wierzchołka v spełniających nasze wymagania (ponieważ mamy do czynienia z drzewem - poddrzewa nie są połączone żadnymi krawędziami, trzeba zająć się jedynie korzeniem).