

rys. 22.6. Graf skierowany z zad. 22.3-2 i 22.5-2.

22.3-2. Zilustruj działanie przeszukiwania w głąb dla grafu z rys. 22.6. Załóż, że w pętli **for** w wierszach 5–7 procedury DFS wierzchołki są przeglądane w porządku alfabetycznym i każda lista sąsiedztwa jest uporządkowana alfabetycznie. Podaj dla każdego wierzchołka czas jego odwiedzenia i przetworzenia oraz sklasyfikuj każdą krawędź.

22.3-3. Podaj strukturę nawiasów przeszukiwania w głąb pokazanego na rys. 22.4.

22.3-4. Pokaż, że krawędź (u, v) jest

- (a) krawędzią drzewową lub krawędzią w przód wtedy i tylko wtedy, gdy $d[u] < d[v] < f[v] < f[u]$;
- (b) krawędzią powrotną wtedy i tylko wtedy, gdy $d[v] < d[u] < f[u] < f[v]$;
- (c) krawędzią poprzeczną wtedy i tylko wtedy, gdy $d[v] < f[v] < d[u] < f[u]$.

22.3-5. Pokaż, że w grafie nieskierowanym sklasyfikowanie krawędzi (u, v) jako krawędzi drzewowej lub krawędzi powrotnej zależy od tego, która z krawędzi (u, v) lub (v, u) jest napotykana jako pierwsza podczas przeszukiwania w głąb, jest równoważne sklasyfikowaniu krawędzi (u, v) zależnie od kolejności typów na liście typów w schemacie klasyfikacji.

22.3-6. Zapisz procedurę DFS nierekurencyjnie, posilkując się stosem.

22.3-7. Podaj kontrprzykład do hipotezy mówiącej, że jeśli istnieje ścieżka od wierzchołka u do v w grafie skierowanym G i jeśli $d[u] < d[v]$ w przeszukiwaniu w głąb grafu G , to wierzchołek v jest potomkiem wierzchołka u w zbudowanym lesie przeszukiwania w głąb.

22.3-8. Podaj kontrprzykład do hipotezy mówiącej, że jeśli istnieje ścieżka od wierzchołka u do v w grafie skierowanym G , to w dowolnym przeszukiwaniu dostaniemy $d[v] \leq f[u]$.

22.3-9. Zmodyfikuj kod procedury przeszukiwania w głąb w taki sposób, żeby została wypisana każda krawędź grafu skierowanego G wraz z jej typem. Pokaż, jakie zmiany są potrzebne (gdyby cokolwiek musiało zostać zmienione), jeżeli G jest nieskierowany.

22.3-10. Wyjaśnij, w jaki sposób wierzchołek u w grafie skierowanym może zostać jedynym wierzchołkiem w drzewie przeszukiwania w głąb nawet wtedy, gdy wierzchołek u ma w G zarówno krawędzie wychodzące, jak i wchodzące.

22.3-11. Pokaż, że przeszukiwanie w głąb może być użyte do znajdowania spójnych składowych w grafie nieskierowanym G oraz że las przeszukiwania w głąb składa się z tylu drzew, ile spójnych składowych ma G . Dokładniej, pokaż, w jaki sposób zmodyfikować przeszukiwanie w głąb, żeby każdemu wierzchołkowi v przypisać całkowitoliczbową etykietę $cc[v]$ z przedziału $[1, k]$ taką, że $cc[u] = cc[v]$ wtedy i tylko wtedy, gdy wierzchołki u oraz v są w tej samej spójnej składowej, a k jest liczbą spójnych składowych grafu G .

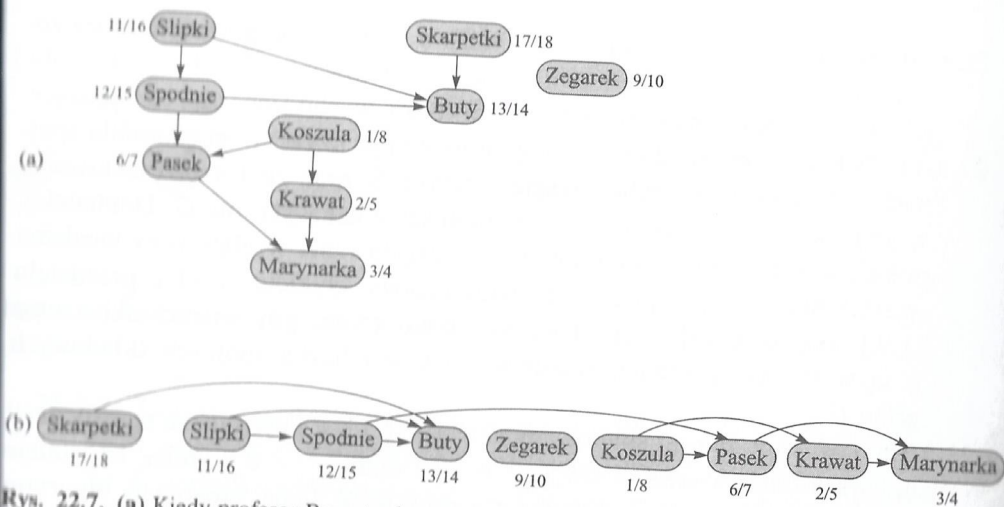
* 22.3-12. Powiemy, że graf skierowany $G = (V, E)$ jest **pojedynczo spójny**, jeśli dla wszystkich wierzchołków $u, v \in V$ z faktu $u \rightsquigarrow v$ wynika, że istnieje co najwyżej jedna prosta ścieżka z u do v w G . Podaj efektywny algorytm sprawdzania, czy dany graf skierowany jest pojedynczo spójny.

22.4. Sortowanie topologiczne

W tym podrozdziale pokażemy, w jaki sposób przeszukiwanie w głąb może być użyte do wykonania sortowania topologicznego acyklicznych grafów skierowanych, zwanych z angielskiego dagami*. **Sortowanie topologiczne** dagu $G = (V, E)$ polega na uporządkowaniu wszystkich jego wierzchołków w taki sposób, że jeśli graf G zawiera krawędź (u, v) , to w tym porządku wierzchołek u występuje przed wierzchołkiem v . (Jeśli graf nie jest acykliczny, to takie uporządkowanie nie jest możliwe). Na sortowanie topologiczne grafu można spojrzeć jak na umieszczanie jego wierzchołków na linii poziomej w taki sposób, żeby wszystkie krawędzie były skierowane od strony lewej do prawej. Sortowanie topologiczne różni się zatem od zwyczajnego sortowania omawianego w części II.

Skierowane grafy acykliczne stosuje się często do określania kolejności wykonywania czynności. Na rysunku 22.7 widać przykładowy graf kolejności wykonywania czynności związanych z porannym ubieraniem się profesora Bumsteada. Profesor musi zakładać pewne części garderoby przed innymi (np. skarpetki przed butami). Inne części garderoby mogą być zakładane w dowolnej kolejności (np. skarpetki i spodnie). W dagu z rys. 22.7(a) krawędź skierowana (u, v) oznacza, że część garderoby u musi zostać założona przed częścią v . Dowolne uporządkowanie topologiczne tego grafu podaje zatem kolejność zakładania składowych ubioru. Na rysunku 22.7(b) widać posortowany topologicznie dag z wierzchoł-

* Dag jest słowem powstałym z pierwszych liter słów *directed acyclic graph*; tym zwyczajnym określeniem będziemy się odąd często posługiwać (przyp. tłum.).



Rys. 22.7. (a) Kiedy profesor Bumstead ma zamiar się ubrać, sortuje topologicznie części swojej garderoby. Każda skierowana krawędź (u, v) oznacza część garderoby u , która musi zostać założona przed częścią v . Obok każdego wierzchołka są pokazane czasy jego odwiedzenia i przetworzenia w przeszukiwaniu w głąb. (b) Ten sam graf posortowany topologicznie. Jego wierzchołki są uporządkowane malejąco względem czasów przetworzenia. Zauważmy, że wszystkie krawędzie skierowane prowadzą od strony lewej do prawej.

kami uporządkowanymi na linii poziomej w taki sposób, że wszystkie krawędzie są skierowane od strony lewej do prawej.

Następujący prosty algorytm służy do sortowania topologicznego danego dagu G .

TOPOLOGICAL-SORT(G)

- 1 wykonaj DFS(G) w celu obliczenia czasów przetworzenia $f[v]$ dla każdego wierzchołka v
- 2 wstaw każdy wierzchołek v na początek listy, kiedy tylko zostanie przetworzony
- 3 **return** lista wierzchołków

Na rysunku 22.7(b) wierzchołki posortowane topologicznie występują w odwrotnym porządku względem ich czasów przetworzenia.

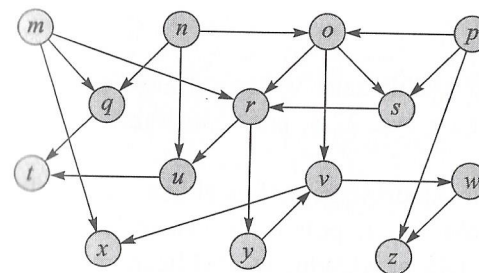
Sortowanie topologiczne można wykonać w czasie $\Theta(V + E)$, ponieważ przeszukiwanie w głąb zabiera czas $\Theta(V + E)$, a czas wstawienia każdego z $|V|$ wierzchołków na początek listy wynosi $O(1)$.

W celu udowodnienia poprawności powyższego algorytmu, użyjemy następującego lematu charakteryzującego acykliczne grafy skierowane.

MAT 22.11

Graf skierowany G jest grafem acyklicznym wtedy i tylko wtedy, gdy przy przeszukiwaniu w głąb grafu G nie pojawiają się krawędzie powrotne.

22.4. SORTOWANIE TOPOLOGICZNE



Rys. 22.8. Dag do ilustracji sortowania topologicznego.

DOWÓD

$\Rightarrow$ Załóżmy, że istnieje krawędź powrotna (u, v) . Wtedy wierzchołek v jest przodkiem wierzchołka u w lesie przeszukiwania w głąb. Zatem w grafie G istnieje ścieżka od v do u , która uzupełniona krawędzią (u, v) daje cykl.

$\Leftarrow$ Załóżmy, że graf G zawiera pewien cykl c . Pokażemy, że w przeszukiwaniu w głąb grafu G pojawia się krawędź powrotna. Niech v będzie pierwszym odwiedzanym wierzchołkiem z cyklu c i niech (u, v) będzie krawędzią poprzedzającą wierzchołek v w cyklu c . W kroku $d[v]$ istnieje ścieżka z v do u zawierająca tylko białe wierzchołki. Z twierdzenia o białej ścieżce wynika, że wierzchołek u staje się potomkiem v w lesie przeszukiwania w głąb. Dlatego (u, v) jest krawędzią powrotną. ♦

TWIERDZENIE 22.12

Algorytm TOPOLOGICAL-SORT(G) sortuje topologicznie acykliczny graf skierowany G .

DOWÓD

Założmy, że algorytm DFS jest wykonywany dla danego dagu $G = (V, E)$ w celu policzenia czasów przetworzenia jego wierzchołków. Wystarczy pokazać, że dla każdej pary różnych wierzchołków $u, v \in V$, jeśli w grafie G istnieje krawędź (u, v) , to $f[v] < f[u]$. Rozważmy krawędź (u, v) badaną w procedurze DFS(G). Kiedy ta krawędź jest badana, wierzchołek v nie może być szary ponieważ wtedy v byłby przodkiem u i krawędź (u, v) byłaby krawędzią powrotną, wbrew lematowi 22.11. Dlatego wierzchołek v musi być albo biały, albo czarny. Jeśli wierzchołek v jest biały, to zostaje potomkiem wierzchołka u , a stąd $f[v] < f[u]$. Jeśli wierzchołek v jest czarny, to znaczy, że był już badany i wartość $f[v]$ została już ustalona. Ponieważ badanie jest nadal prowadzone z u , jeszcze nie została ustalona wartość $f[u]$, a jak to już zostanie zrobione, będziemy mieli $f[v] < f[u]$. Stąd dla każdej krawędzi (u, v) w dagu mamy $f[v] < f[u]$, co kończy dowód twierdzenia. ♦