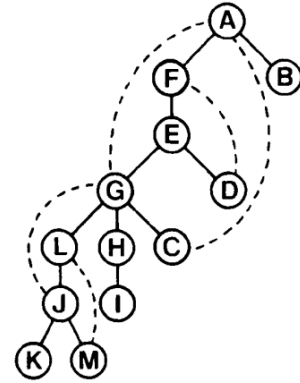


## Анализ на задача „Критична точка“

Критичен връх (артикуляционна точка – точка на съчленяване) е такъв връх, че премахването му води до увеличаване броя на свързаните компоненти. Ако графът е бил свързан преди премахването на върха, той ще бъде несвързан след това.

Намирането на критичните точки може да стане чрез DFS и използване на правите и обратните ребра

В примера на рисунката изтриването на възел E няма да прекъсне връзката с графа, защото G и D имат пунктирани връзки (обратни ребра при DFS), които сочат над E, давайки алтернативни пътища от тях до F (бащата на E в дървото). Но, изтриването на G ще прекъсне връзката с графика, защото няма такива алтернативни пътища от L или H до E (бащата на L).



Така виждаме, че един връх  $x$  НЕ E критичен, ако за всеки негов син  $y$  има някакъв връх по-надолу в поддървото на  $y$ , свързан (чрез пунктирана връзка, т.е. чрез обратно ребро) с връх, по-нагоре от  $x$  в дървото (така се осигурява допълнителна връзка от  $x$  към  $y$ ).

Обаче, гореописаното не работи, когато  $x$  е корена на дървото, получено от DFS, тъй като няма възли „по-високо в дървото“. В този случай трябва да се направи отделна проверка за броя на децата на  $x$  в дървото, получено от DFS, защото, както лесно се вижда, коренът е критичен връх, тогава само тогава, когато той има два или повече синове, тъй като единственият път, свързващ синовете на корена, минава през корена.

**1. Наивен подход за решаване на задачата:** Премахваме поотделно всеки връх (заедно с инцидентните му ребра) и проверяваме дали полученият граф е свързан. Това е реализирано в програмата `crit_40p.cpp`.

**2. По-добър начин за решаване на задачата:** За всеки връх да се направи проверка чрез обратните ребра. Това е реализирано в програмата `crit_80p.cpp`, където във векторът `rf` се пресмятат правите ребра, а в променливата `c0` – броя на децата на корена. Функцията `find_rr()` намира обратните ребра. Във функцията `find_AP()`, чрез цикъла `for (int v=0; v<V; v++)` се проверява чрез извикване на `dfs()` за всеки връх дали е критична точка, като се изследват поддърветата на неговите синове.

**3. Решение за 100 точки.** Програмата за намиране на критични върхове може да се направи още по-добра, като се извършва само едно търсене в дълбочина. Това е реализирано в `crit_100p.cpp`. Там в модифицираната функция за търсене в дълбочина `int mdfs(int v)` се връща стойността на променливата `min`, в която за всеки връх  $v$  се пресмята най-малкият нов номер при DFS, към който сочат върхове от поддърветата на синовете на върха  $v$ . Функцията `dfs0(int v)` служи за намиране броя на синовете на корена.