Problema Arbore –
autor Denis-Gabriel Mită, Universitatea din București

Soluție O(2^N * N) - 20

Ne setăm fiecare submulțime posibilă de noduri ce vrem să le eliminăm și verificăm numărul de componente conexe cu o parcurgere în adâncime / lățime. Complexitatea finală este O(2^N * N).

Soluție O(N) prima cerință - 40

Pentru a afla numarul maxim de componente conexe putem face observația că la orice pas nu are rost sa ștergem o frunză, deoarece numărul de componente conexe nu va crește. În schimb, dacă am elimina vecinul lor, toate frunzele legate de acesta vor forma componente conexe independente. Dacă nu l-am elimina, în cazul în care acesta ar avea legate cel puțin 2 frunze, eliminându-l ne-ar crește numărul de componente conexe, iar neeliminându-l numărul de componente conexe nu ar putea crește față de cazul în care l-am elimina. Astfel, introducem nodurile cu gradul 1 într-o coadă, iar la fiecare pas scoatem din coadă un nod cu grad 1, dacă nu este șters incrementăm numărul maxim de componente conexe, iar apoi ștergem vecinul său și reintroducem in coadă nodurile nou create cu grad 1. Complexitatea finală este O(N).

Soluție O(N) - 100

Ne fixăm arborele într-o rădăcină (spre exemplu nodul 1) și apoi ne construim următoarea dinamică printr-o parcurgere în adâncime:

nr[0][nod] = numărul maxim de componente conexe rezultate din subarborele nodului nod dacă NU eliminăm nodul nod

nr[1][nod] = numărul maxim de componente conexe rezultate din subarborele nodului nod dacă eliminăm nodul nod

dp[0][nod] = numărul de moduri în care obținem nr[0][nod] componente conexe din subarborele nodului nod dacă NU eliminăm nodul nod

dp[1][nod] = numărul de moduri în care obținem nr[1][nod] componente conexe din subarborele nodului nod dacă eliminăm nodul nod

Recurența este următoarea:

nr[0][nod] = 1 + sumă din max(nr[1][fiu], nr[0][fiu] - 1)
nr[1][nod] = sumă din max(nr[1][fiu], nr[0][fiu])
dp[0][nod] = 1 * produs din : dp[1][fiu] + dp[0][fiu] dacă nr[1][fiu] = nr[0][fiu] - 1
dp[1][fiu] dacă nr[1][fiu] > nr[0][fiu] - 1
dp[0][fiu] dacă nr[1][fiu] < nr[0][fiu] - 1
dp[1][nod] = 1 * produs din : dp[1][fiu] + dp[0][fiu] dacă nr[1][fiu] = nr[0][fiu]
dp[1][fiu] dacă nr[1][fiu] > nr[0][fiu]
dp[0][fiu] dacă nr[1][fiu] < nr[0][fiu]

Soluția pentru prima cerință se va fi max(nr[0][1], nr[1][1]), iar pentru a doua cerință va fi dp[0][1] dacă nr[0][1] > nr[1][1], în dp[1][1] dacă nr[1][1] > nr[0][1], sau dp[0][1] + dp[1][1] dacă nr[0][1] = nr[1][1]. Complexitatea finală este O(N).