Tabăra de pregătire a Lotului Naţional de Informatică
Focşani, 15-22 mai, 2010
Baraj 3 Seniori

Solutie Ubergraf

Autori: Paul Baltescu, Alexandru Tomescu

Problema se rezolva cu metoda programarii dinamice. O prima observatie este ca daca unui *ubergraf* cu N noduri ii eliminam un nod cu grad intern 0, graful ramas este un *ubergraf* cu N-1 noduri. Vom incerca sa construim o relatie de recurenta plecand de la aceasta observatie. Observam totusi ca daca incercam sa construim astfel relatia de recurenta, vom numara anumite *ubergraf*-uri de mai multe ori.

Solutia este sa construim *ubergraf*-urile "pe nivele". Cunoastem faptul ca un *ubergraf* are un singur nod cu grad extern 0, caruia ii atribuim inaltimea 0. Pentru restul nodurilor definim inaltimea ca numarul maxim de arce ce trebuie parcurs de la nodul respectiv la nodul cu grad extern 0. Definim matricea C[i][j] = numarul de *ubergraf*-uri cu i noduri, dintre care j se afla pe nivelul maxim, cu recurenta ascociata $C[i][j] = \Sigma\ C[i-j][k] * cmb(2^{i-j} - 2^{i-j-k}, j)$. (Prin cmb(n, k) am notat combinari de n luate cate k). In continuare vom explica cum s-a ajuns la aceasta recurenta. Sa presupunem ca dorim sa numaram cate *ubergraf*-uri exista avand i noduri dintre care j pe ultimul nivel. Pe nivelul anterior presupunem ca exista un numar oarecare de noduri k. Trebuie sa asociem pentru fiecare din cele j noduri, o multime de vecini distincta, cel putin unul dintre vecini aflandu-se pe nivelul anterior. Numarul de multimi de vecini se calculeaza scazand din numarul total de posibilitati 2^{i-j} cele care nu contin un nod pe penultimul nivel 2^{i-j-k}.

Daca nu luam in calcul timpul necesar calculului combinarilor, recurenta gasita are complexitate $O(N^3)$. Observam ca unul dintre "termenii" combinarilor este j care va fi intotdeauna mai mic decat N. Fie m un numar oarecare. Putem scrie cmb(m, j) = (m-j+1)*...*m / j!, fractie care poate fi calculata in complexitate O(N) datorita observatiei anterioare (atat numaratorul, cat si numitorul au maxim N factori). Asadar, aceasta rezolvare ar avea complexitate $O(N^4)$ si ar obtine 65 de puncte.

Recurenta poate fi redusa la $O(N^3)$ pe urmatoarea observatie: $2^{i-j}-2^{i-j-k} = 2^{i+1-(j+1)} - 2^{i+1-(j+1)-k} \Rightarrow cmb(2^{i+1-(j+1)} - 2^{i+1-(j+1)-k}, j) = cmb(2^{i-j}-2^{i-j-k}, j)$. Cum noi vom avea nevoie de $cmb(2^{i+1-(j+1)} - 2^{i+1-(j+1)-k}, j+1)$, putem obtine aceasta valoare in timp O(1) astfel: $cmb(2^{i+1-(j+1)} - 2^{i+1-(j+1)-k}, j+1) = cmb(2^{i-j}-2^{i-j-k}, j) * (2^{i+1-(j+1)} - 2^{i+1-(j+1)-k}-j) / (j+1)$.

Pentru a putea calcula recurenta de mai sus este necesar sa precalculam puterile lui 2, pentru exponentii cuprinsi intre 0 si N si inversii modulari ai numerelor {1...N} fata de P. Prima precalculare se poate face fara probleme in complexitate O(N), iar a doua precalculare este suficient sa se realizeze in complexitate O(N*P), pentru fiecare numar din multimea {1...N} iterand prin valorile {1...P} si gasind numarul care verifica conditia de invers modular. Mentionam faptul ca inversul modular al unui numar poate fi calculat si in complexitate O(logP) folosind Algoritmul lui Euclid extins sau Mica Teorema a lui Fermat, dar acest lucru nu este necesar. Asadar, complexitatea solutiei oficiale este $O(N*P + N^3)$.