Problema 2 – ubuntzei

autor stud. Marius Stroe
Universitatea „Babeș Bolyai”, Facultatea de Informatică
Cluj-Napoca

Soluție 100 puncte

Fie `G = (V, E)` graful neorientat ce modelează țara din enunțul problemei. Cerința constă în a găsi un drum de lungime minimă în graful `G` de la nodul de start la nodul destinație trecând cel puțin o dată prin cele `K` orașe date.

În acest sens, construim un nou graf `G′ = (V′, E′)`, unde `V′` este o mulțime de perechi `(u, s)`, astfel încât `u` este un nod din `V`, iar `s` este o submulțime a celor `K` orașe. Perechea `(u, s)` semnifică că suntem în nodul `u` și am vizitat nodurile din mulțimea `s`.

Astfel, vom aplica algoritmul lui *Dijkstra* pe acest graf căutând drumul de lungime minimă de la nodul `(1, {})` la nodul $(N, \{C_1, C_2, .. C_K\})$. În implementare, mulțimile se pot reține cu ajutorul biților și se pot manipula cu ajutorul măștilor pe biți.

Complexitate finală: $O(K * M \log_2(N) + 2^K K^2 \log_2(K))$.

Se observă că graful *G′* este aciclic. Astfel, o soluție alternativă folosește metoda *programării dinamice*, calculând `dp(i, s)` ca fiind costul minim de a ajunge în orașul `i` și să fi trecut cel puțin o dată prin orașele din mulțimea `s`. Recurența este

$$dp(i, s) = \min \{ dp(j, s - \{i\}) + dist_G(i, j) : j \text{ în } s \text{ și } j \neq i \}$$

unde $dist_G(i, j)$ reprezintă distanța în graful `G`, graful inițial, de la nodul `i` la nodul `j`.

Soluție 70 de puncte

Această soluție se bazează pe ideea anterioară, doar că drumurile minime dintre oricare două noduri din cele `K` se calculează cu algoritmul lui *Floyd-Warshall* în complexitate $O(N^3)$. Iar drumul minim în graful `G′` se calculează cu oricare din variantele prezentate la soluția anterioară, inclusiv *algoritmul lui Bellman Ford* cu coadă.

Soluție 50 puncte

Vom precalcula cu algoritmul lui *Floyd-Warshall* distanțele dintre oricare două noduri din graful `G`. Fie `d` matricea distanțelor obținută la terminarea algoritmului.

Ulterior, enumerăm toate permutările `P` ale celor `K` noduri, calculăm pentru fiecare distanța $d(1, P_1) + d(P_1, P_2) + ... + d(P_{K-1}, P_K) + d(P_K, N)$ și alegem cea mai mică valoare, ce va fi răspunsul.

Complexitate finală: $O(N^3 + K!)$.

Soluție 20 de puncte

Se va calcula lungimea drumului minim de la nodul `1` la nodul `N`, utilizând orice algoritm de drum minim cunoscut.