



MinDAG (mindag)

Author: Bogdan-Ioan Popa

Developer: Bogdan-Ioan Popa

Solution

Fie $C_{i,j}$ costul pentru a orienta muchia (i, j) de la i la j . Problema cere de fapt gasirea costului minim generat de o permutare P conform formulei $\sum_{1 \leq i < j < N} C_{P_i, P_j}$.

Putem folosi programare dinamica pentru a gasi acest cost minim.

$$dp_{mask} = \min_{i \in mask} (dp_{mask \setminus \{i\}} + Cost(mask \setminus \{i\}, i))$$

unde $mask$ reprezinta o submultime de noduri, iar

$$Cost(mask, i) = \sum_{j \in mask} C_{j,i}$$

Pentru a obine 100 de puncte, trebuie sa precalculam $Cost(mask, i)$, printr-o tehnic asemntoare cu meet in the middle.

Vom precalcula $Cost(mask, i)$ pentru $mask \subseteq \{1, 2, \dots, \frac{N}{2}\}$ si pentru $mask \subseteq \{\frac{N}{2}+1, \dots, N\}$ in $O(2^{\frac{N}{2}} * N)$.

Complexitatea timp final este $O(2^N * N)$, iar complexitatea memorie este $O(2^N)$.

Let $C_{i,j}$ be the cost of orienting the edge (i, j) from i to j . The problem essentially requires finding the minimum cost generated by a permutation P according to the formula:

$$\sum_{1 \leq i < j < N} C_{P_i, P_j}$$

We can use dynamic programming to find this minimum cost:

$$dp_{mask} = \min_{i \in mask} (dp_{mask \setminus \{i\}} + Cost(mask \setminus \{i\}, i))$$

where $mask$ represents a subset of nodes, and

$$Cost(mask, i) = \sum_{j \in mask} C_{j,i}$$

To achieve a full score of 100 points, we need to precompute $Cost(mask, i)$ using a technique similar to meet-in-the-middle.

We will precompute $Cost(mask, i)$ for $mask \subseteq \{1, 2, \dots, \frac{N}{2}\}$ and for $mask \subseteq \{\frac{N}{2}+1, \dots, N\}$ in $O(2^{\frac{N}{2}} \cdot N)$.

The final time complexity is $O(2^N \cdot N)$, and the memory complexity is $O(2^N)$.



SD (SD)

Author: Alin-Gabriel Raileanu, Alexandru Gheorghies

Developer: Alin-Gabriel Raileanu, Alexandru Gheorghies

Solution

Initial, daca avem un sir vazut ca circular, secventa de suma maxima din acest sir va fi ori secventa maxima din sirul necircular (1), ori secventa formata de un prefix si un sufix care nu se suprapun din sirul necircular (2).

Pentru (2) se poate observa ca prefixul si sufixul din alegerea optima sunt cele obtinute prin eliminarea secventei de suma minima.

Asadar, pentru rezolvarea problemei este suficienta retinerea unui arbore de intervale menit sa poata rezolva urmatoarele operatii:

- Actualizare: schimbarea valorii corespondente unei pozitii;
- Interogare: suma maxima a unei secvente pe interval; (*)
- Interogare: suma minim a unei secvente pe interval; (**)
- Interogare: suma pe interval. (***)

Raspunsul pentru fiecare operatie de tipul 2 va fi maximul dintre (*) si (***)-(**).

Complexitate timp: $O(N + Q \cdot \log(N))$.

Complexitate spatiu: $O(N)$.