АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА МАКСИМАЛНО КОНСЕНСУСНО ДЪРВО

Намирането на маскималното консенсусно дърво на две двоични филогенетични дървета може да бъде реализирано за $\theta(n^2)$* памет и време, използвайки динамическо програмиране. Състоянията, от които се интересуваме представяват двойки укоренени двоични поддървета *(subP, subQ)* в дървета *P* и *Q*. Всяко поддърво представлява половината от първоначалното дърво, „отрязана“ по фиксирано насочено ребро – края на реброто указва корена на поддървото, а началото на реброто указва върха, който не се включва в поддървото.

Максимизираме функцията `dp(subP, subQ)` – броя на листата в консенсусното дърво на двете поддървета по следните правила:

- Ако *subP* и *subQ* включват само по едно листо (съответно, *leafP* и *leafQ*), то:

  `dp(subP, subQ) := is_equal(leafP, leafQ),`

  където `is_equal` връща 1, ако имената на двете листа са еднакви и 0 в противен случай;;

- Ако и двете поддървета *subP* и *subQ* имат по две деца (т.е. включват по повече от едно листо):

```
dp(subP, subQ) := max{ dp(LP, LQ) + dp(RP, RQ),
           dp(LP, RQ) + dp(RP, LQ) },
```

където *LP* и *RP* са двете дъщерни поддървета на *P*, а *LQ* и *RQ* са двете дъщерни поддървета на *Q*;

- Ако точно едно от поддървета *subP* и *subQ* съдържа само едно листо (без ограничение на общността, нека това е *subQ*), то:

```
dp(subP, subQ) := max{ dp(LP, subQ), dp(RP, subQ) },
```

където *LP* и *RP* са двете дъщерни поддървета на *P*.

Очевидно, при вече изчислени функциите dp в дясната част на уравненията, изчисляването на лявата част изисква константни време и памет.

Така функцията *dp* дава отговор за всички възможни съчетания на укоренени поддървета. Но в задачата се изисква масимизация по неукоренените дървета *P* и *Q*. За допълнително време $\theta(n^2)$ можем да разрежем всяко от дърветата *P* и *Q* по произволни ребра на по две допълващи се поддървета *subP1□subP2=subP* и *subQ1□subQ2=subQ*. Отговор на задачата е максималната сума *dp(subP1, subQ1) + dp(subP2, subP2)* по всички наредена двойки ребра, по които „разрязваме“ дърветата. Описаното дотук решение е достатъчно за набиране на пълен брой точки за задачата.

Възможно е частично решение, изпробващо всички подмножества от листа на едното дърво. Такова решение има сложност по време $\Omega(2^n)$** и се очаква да получи около x% от точките за задачата.

Друго възможно частично решение, използващо жаден подход, последователно уголемяващ консенсусното дърво:

1. Избираме произволно листо, срещащо се и в двете дървета;
2. Ако съществува такова листо, което все още не е включено в консенсусното дърво и няма да наруши консенсусното свойство, то включваме това листо;
3. Ако не можем повече да уголемим консенсусното дърво, но имаме на разположение още процесорно време, започваме отначало точка 1.

В зависимост от това, в какъв ред добавяме листата в точка 2. зависи дали ще достигнем до максимално консенсусно дърво. Ако максималното консенсусно дърво съдържа *L* листа, то всяка от $L_t!$ (т.е. *1*2*...*Lbest*) последователности, започващи с листата на максимално консенсусно дърво, ще доведе до намиране на правилния отговор.

* *θ (тета) обозначава точно асимптотично ограничение (с точност до умножение на константа)*

** *Ω (омега) обозначава асимптотично ограничение отдолу (с точност до умножение на константа)*

Автор: Петър Иванов
Автор на жадното решение: Искрен Чернев