Erinaceida

2022, Baraj 1, Seniori

Problema numărului maxim de fructe se reduce la găsirea celui mai lung subșir crescător. Pentru fiecare punct, vom ține minte numărul maxim de fructe culese până în acel fruct (practic, lungimea celui mai lung subșir crescător care conține fructul respectiv) și un rucsac care ne spune pentru fiecare cantitate de fructe dintr-un anumit tip dacă se poate obține, deci valoarea:

```
sePoate[N][M] = true dacă există un drum până la fructul N care
conține
                un număr maximal de fructe și exact M mere;
```

Deoarece această dinamică conține doar biți, putem folosi un bitset pentru a avea o constantă de 1 / 64.

Dacă pentru un fruct, numărul maxim de fructe culese este G, atunci bitsetul acelui fruct va fi:

```
sePoate[N] = or(sePoate[i] << 1 unde i este un fruct de la care se
               poate ajunge în N, iar numărul maxim de fructe culese
               în i este G - 1).
```

Această soluție are complexitate O(N^3 / 64): calculăm dinamica pentru N fructe, pentru un fruct, avem N termeni, iar pentru a face un or pe bitseturi, avem nevoie de N / 64 operații.

Există mai multe observații care duc la soluția de 100 de puncte:

- Pentru fiecare G posibil, fructele pentru care se obține acel G formează niște diagonale;
- Un fruct de pe diagonala G își va lua termenii din recurență dintr-un interval compact de pe diagonala G - 1. Dacă rezolvăm fructele de pe o diagonală G în ordine, atunci termenii incluși de pe diagonala G - 1 simulează un two-pointers;
- Avem nevoie de o coadă care calculează or-ul tuturor bitseturilor incluse;
- Dacă ai calculată suma or-urilor, e greu să scoți unul din termeni;
- O coadă poate fi simulată cu două stive, iar pentru fiecare termen din stivă, calculăm suma or a tuturor termenilor de sub acel termen.

Astfel, soluția se amortizează la O(N^2 / 64): fiecare bitset este adăugat/scos o singură dată într-o stivă în O(N / 64), iar pentru a combina două stive pentru fiecare fruct, se rezolvă tot în O(N / 64).