

Greedy

BUZATU GIULIAN & NIȚĂ ALEXANDROS

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/353/spectacole>

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/353/spectacole>

Hint: După care din următoarele criterii de sortare obținem optimul?

- a) Durata spectacolelor
- b) Ora de început
- c) Ora de sfârșit

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/353/spectacole>

Soluție: Observăm că modalitatea optimă de sortare este cea de la subpunctul c). Astfel, sortăm spectacolele după ora de sfârșit. După aceea alegem primul spectacol, iar pentru fiecare spectacol rămas verificăm dacă începe după ultimul spectacol ales, caz în care îl adăugăm și pe el.

Complexitate: $O(n \log n)$

Implementare: <https://github.com/Giulian617/Hai-la-olimpiada-2023-2024/blob/main/11-12/resources/spectacole.cpp>

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/1340/rucsac>

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/1340/rucsac>

Hint: Cum putem sorta pentru a obține un profit optim?

Problemă

Să rezolvăm problema: <https://pbinfo.ro/probleme/1340/rucsac>

Soluție: Intuitiv, pentru a obține un câștig maxim, trebuie să încărcăm mai întâi în rucsac obiectele care sunt cele mai eficiente, deci sortăm descrescător după raportul dintre cost și greutate. Parcurgem obiectele sortate și avem două cazuri: obiectul încapă complet în rucsac, deci îl adăugăm complet, sau nu este loc suficient pentru obiect, deci îl adăugăm fracționar și ne oprim. Putem să încetăm parcurgerea, deoarece știm că după ce luăm un obiect fracționar, nu mai avem loc în rucsac pentru altele. În momentul adăugării unui obiect în rucsac, actualizăm răspunsul.

Implementare: <https://github.com/Giulian617/Hai-la-olimpiada-2023-2024/blob/main/11-12/resources/rucsac.cpp>

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/elmaj>

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/elmaj>

Hint 1: Putem începe prin găsirea unei soluții mai slabe.

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/elmaj>

Hint 1: Putem începe prin găsirea unei soluții mai slabe.

Hint 2: Pentru soluția optimă, încercați să vă gândiți la un proces prin care rămânem cu un candidat care poate fi element majoritar.

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/elmaj>

Soluție: Vom crea perechi cu elemente diferite. Dacă la final rămân elemente care nu au fost adăugate în astfel de perechi, atunci avem un candidat. Pentru formarea perechilor, vom folosi totuși o abordare abstractă, folosind o variabilă contor. La final, trebuie să verificăm dacă candidatul nostru este într-adevăr element majoritar.

Complexitate: $O(n)$

Implementare: https://infoarena.ro/job_detail/3197465?action=view-source

Problemă

Să rezolvăm problema: <https://codeforces.com/problemset/problem/1907/C>

Problemă

Să rezolvăm problema: <https://codeforces.com/problemset/problem/1907/C>

Hint: Cum putem reduce problema la o problemă de greedy cunoscută?

Problemă

Să rezolvăm problema: <https://codeforces.com/problemset/problem/1907/C>

Soluție: Problema se reduce la găsirea unui element majoritar. Dacă avem un element majoritar, atunci, la fiecare pas, o să putem face o eliminare care conține o apariție a elementului majoritar și un alt element din string-ul nostru. Astfel, la final, vom rămâne doar cu elemente de tipul celui care era majoritar la început. Dacă nu aveam un element majoritar, atunci putem elimina toate literele dacă lungimea string-ului este pară, sau rămânem cu o singură literă, dacă lungimea este impară.

Implementare: <https://codeforces.com/contest/1907/submission/236777268>

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/linterv>

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/linterv>

Hint: Căutăm un mod optim de a obține intervale disjuncte.

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/linterv>

Soluție: Sortăm intervalele crescător după capătul stânga și, în caz de egalitate, crescător după capătul dreapta. Parcurgem intervalele, iar dacă intervalul curent se intersectează sau este inclus în cel precedent, îi modificăm capetele ca cele 2 să se mai intersecteze doar în capătul dreapta al celui precedent (aspect care nu ne afectează soluția). După ce modificăm intervalul actual, adăugăm la răspuns diferența dintre cele două capete.

Implementare: https://infoarena.ro/job_detail/3197501?action=view-source

Temă

- <https://pbinfo.ro/probleme/950/cerc3>
- <https://pbinfo.ro/probleme/2683/easy-ssc>
- <https://pbinfo.ro/probleme/2684/hard-ssc>
- <https://pbinfo.ro/probleme/400/pachete>
- <https://pbinfo.ro/probleme/401/pachete-multe>
- <https://pbinfo.ro/probleme/1425/ghirlande>

Probleme suplimentare

- <https://infoarena.ro/problema/lacate>
- <https://infoarena.ro/problema/cai>
- <https://codeforces.com/problemset/problem/1353/E>

Lectură suplimentară

- <https://github.com/Giulian617/Hai-la-olimpiada-2023-2024/blob/main/11-12/resources/Lectie-Greedy-Radu-Boriga.pdf> - Curs predat de domnul profesor dr. Radu Boriga, în cadrul unui curs pe care dumnealui îl ține la "Facultatea de Matematică și Informatică", din cadrul "Universității din București"
- <https://infoarena.ro/problema-majoritatii-votului>