Algoritmi Avansați
Seminar 1

Gabriel Majeri

Pentru fiecare dintre problemele de mai jos, propuneți o **soluție** care să fie cât mai **eficientă** (i.e. să respecte complexitatea indicată) și **demonstrați** / **justificați** formal corectitudinea rezolvării voastre.

Acoperirea cu intervale

Descrierea problemei. Suntem administratorii unui spital și vrem să ne asigurăm că avem tot timpul cel puțin un medic prezent la camera de gardă, într-un anumit interval orar. Presupunem că avem în subordine n medici, fiecare fiind dispus să stea de gardă între anumite ore. Vrem să-i asignăm pe cât mai puțini dintre ei la serviciul de gardă, lăsându-i pe ceilalți să se ocupe de restul pacienților.

Formularea matematică. Find dat un interval $[a, b]$ și o mulțime de intervale $[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]$, să se aleagă un număr *minim* dintre acestea astfel încât, reunite, să includă intervalul inițial $[a, b]$.

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \log n)$

Problema rucsacului

Descrierea problemei. Se dă o mulțime de obiecte $O = \{ o_1, o_2, \dots, o_n \}$. Pentru fiecare obiect o_i știm cât este **valoarea** obiectului (pe care o vom nota val_i), respectiv **greutatea** obiectului (pe care o vom nota g_i).

Scopul este să găsim o submulțime de obiecte $S \subseteq O$ astfel încât suma greutăților obiectelor din S să nu depășească o capacitate C a unui rucsac (fixată de la început), iar valoarea totală a acestora să fie maximă.

Formularea matematică. Să se găsească $S \subseteq O$ care să maximizeze

$$\sum_{o_i \in S} val_i$$

cu condiția că

$$\sum_{o_i \in S} g_i \leq C$$

Observație. Toate submulțimile S care respectă condiția de mai sus se vor numi **soluții candidat**. Dintre acestea, soluția (sau soluțiile) care maximizează valoarea totală se va numi **soluția optimă**.

Varianta fracționară

Elementele din S pot fi fracțiuni din obiecte; cu alte cuvinte, pentru fiecare obiect o_i reținem și procentul din el pe care îl luăm în considerare, $p_i \in [0, 1]$. Fiecare obiect din soluție contribuie la valoarea totală cu $p_i \cdot val_i$, respectiv este contorizat spre greutatea totală cu $p_i \cdot g_i$.

Formularea matematică devine: să se găsească $S \subseteq O$ care să maximizeze $\sum_{o_i \in O} p_i \cdot val_i$ cu condiția că $\sum_{o_i \in O} p_i \cdot g_i \leq C$.

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \log n)$

Varianta discretă

Elementele din S pot fi doar obiecte luate cu totul; cu alte cuvinte, pentru fiecare obiect o_i decidem dacă îl includem sau nu.

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \cdot C)$

Alte probleme care se rezolvă în timp pseudo-polinomial

Submulțime de elemente cu suma dată

Se dă o mulțime de n numere naturale și un număr natural $M < 10000$. Să se determine, dacă există, o submulțime a mulțimii date de sumă egală cu M.

Exemplu. Pentru $n = 6$, mulțimea $\{ 12, 1, 3, 4, 5, 7 \}$ și $M = 14$, o soluție este submulțimea $\{ 3, 4, 7 \}$.

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \cdot M)$

Împărțirea cadourilor

Moș Crăciun a poposit la bradul a doi frați, unde și-a golit sacul. Când s-au trezit, frații au intrat într-o mare dilemă: cum își vor împărți ei cadourile? Știind că fiecare cadou are o valoare cuprinsă între 1 și 100 și că sunt maxim 100 de cadouri, scrieți un program care să determine sumele cadourilor fraților precum și modul de împărțire, astfel încât sumele obținute să fie cât de apropiate posibil.

Exemplu. Pentru 7 cadouri cu valorile $\{28, 7, 11, 8, 9, 7, 27\}$, sumele sunt 48 și 49, o împărțire a cadourilor fiind: $\{28, 11, 9\}$, respectiv $\{7, 8, 7, 27\}$.

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \cdot S)$, unde S reprezintă valoarea totală a cadourilor.

Alte probleme de programare dinamică

Generalizarea problemei spectacolelor

Se dau n activități, pentru fiecare având: **timpul de început**, **timpul de sfârșit** și **profitul** asociat desfășurării activității (adică n intervale de numere reale, fiecare cu o pondere asociată). Să se determine o submulțime de activități compatibile (intervale disjuncte două câte două) care au profitul total maxim. Se vor afișa profitul total și activitățile selectate.

Exemplu. Pentru $n = 4$ și activitățile:

- $A_1 = [1, 3]$ cu profitul $p_1 = 1$
- $A_2 = [2, 6]$ cu profit $p_2 = 8$
- $A_3 = [4, 7]$ cu profit $p_3 = 2$
- $A_4 = [10, 15]$ cu profit $p_4 = 5$

O soluție optimă ar fi formată din activitățile A_2 și A_4 (adică intervalele $[2, 6]$ și $[10, 15]$), cu profitul total 13.

Complexitatea pe care ar trebui să o aibă soluția voastră: încercați să găsiți o soluție în $\mathcal{O}(n \log n)$; dacă nu reușiți, încadrați-vă în cel mult $\mathcal{O}(n^2)$.

Generalizarea problemei planificării activităților

Se dă o listă de n activități. Pentru fiecare activitate cunoaștem **profitul** (p_i), **termenul limită** până la care trebuie efectuată (t_i) și **durata** acesteia

(l_i). Trebuie să găsim o submulțime *ordonată* (o sublistă) de activități pentru a obține un profit total maxim.

Exemplu. Pentru $n = 4$ și activitățile:

- A_1 cu $p_1 = 3$, $t_1 = 5$, $l_1 = 3$
- A_2 cu $p_2 = 2$, $t_2 = 2$, $l_2 = 1$
- A_3 cu $p_3 = 3$, $t_3 = 2$, $l_3 = 2$
- A_4 cu $p_4 = 5$, $t_4 = 4$, $l_4 = 3$

O soluție optimă se obține dacă planificăm activitățile în ordinea A_2, A_4, profitul fiind 7.

Observație. Problema discretă a rucsacului poate fi privită ca un caz particular al acestei probleme (obiectele sunt activități de durată g_i, profit c_i și termen limită C).

Complexitatea pe care ar trebui să o aibă soluția voastră: $\mathcal{O}(n \cdot T + n \log n)$, unde $T = \max \{ t_i \}$.