PROBLEMA 1 Electoral

Autor: Student Andrei Netedu, Facultatea de Informatică, Universitatea Al. I. Cuza Iaşi

Soluţii – stud. Andrei Netedu, stud. Valentin Rosca, stud. Cristian Vîntur

Considerăm şirul de candidaţi care participă in faza a doua a alegerilor. Fie x elementul majoritar din acest şir. In urma schimbarilor unor voturi, acest sir poate fi modificat prin 3 tipuri de operatii, in functie de rezultatele din grupa:

- 1. adaugare: intr-un district in care nu a iesit niciun candidat majoritar, facem astfel incat sa iasa un candidat majoritar, evident diferit de x.
- 2. eliminare: intr-un district in care a iesit x candidat majoritar, facem astfel incat sa nu mai existe niciun candidat majoritar
- 3. schimbare: asemanator cu eliminarea, doar ca in loc sa nu iasa nimeni majoritar, sa iasa altcineva

Se observa ca nu exista un district in care sa putem folosi toate operatiile. De asemenea se observa ca pentru fiecare district este suficient sa aplicam maxim o operatie din cele 3.

Pentru fiecare district si fiecare operatie exista un cost asociat aplicarii operatii in districtul respectiv. Notam acest cost cu `c(d, o), 1<=d<=n, 1<=o<=3.`

Consideram acum functia `f(s) = cnt(s, x) – cnt(s, y | y != x)` unde s este un sir, iar x este elementul care apare de cele mai multe ori in s (sau unul dintre ele, daca sunt mai multe).
`cnt(s, x)` = numarul de aparitii ale lui x in s
`cnt(s, y | y != x)` = suma numarului de aparitii ale tuturor numerelor diferite de x care apar in s

Se observa ca daca `f(s) > 0` atunci x este element majoritar, iar daca `f(s) <= 0` atunci nu exista element majoritar.
In urma aplicarii unei operatii dintre cele de mai sus pe un sir s, obtinandu-se sirul s', valoarea functiei f se schimba `(f(s) != f(s'))`. Se arata ca in urma unei adaugari sau eliminari, f(s') = f(s) – 1, iar in urma unei schimbari `f(s') = f(s) – 2`.

Pentru fiecare district asociem 2 costuri:
costul pentru a scadea functia cu 1 (prin adaugare / eliminare) : c_i^1 pentru districtul i
costul pentru a scadea functia cu 2 (prin schimbare) : c_i^2 ($c_i^2 = INF$ daca operatia nu este valida)

Problema se reduce la urmatoarea: avem n grupe, fiecare grupa avand o pereche de obiecte (a_i^1, a_i^2). a_i^1 are costul c_i^1 si valoarea 1, iar a_i^2 are costul c_i^2 si valoarea 2. Din fiecare grupa trebuie sa alegem maxim un element din pereche, astfel incat suma valorilor obiectelor alese sa fie mai mare sau egala cu f(s) (s fiind sirul initial), iar suma costurilor sa fie minima.

Solutia 40 de puncte (doar pe a doua cerinta 60 de puncte in total)
Utilizam o dinamica de tip rucsac: `dp[i][j]` = costul minim astfel incat, alegand din primele i obiecte, sa avem valoarea exact j.
Recurenta este: `dp[i][j] = min(dp[i – 1][j], dp[i – 1][j – 1]+`c_i^1`, dp[i – 1][j – 2]+`c_i^2`)`
Raspunsul se afla in `dp[n][f(s)]` Deoarece recurenta foloseste doar valori de pe linia anterioara, este suficienta o matrice cu 2 linii si n coloane.

Solutia 100 de puncte.

Din cele n perechi de doua obiecte de greutate 1 si 2, este necesar sa gasim o multime de obiecte care au greutatea insumata exact g.

Din fiecare set de doua obiecte vom alege maxim un obiect in solutia noastra.
Fie c_i^1 costul pentru a elimina sau adauga un element majoritar, si c_i^2 costul pentru a schimba un element majoritar.
Stim ca $c_i^1 <= c_i^2$ pentru oricare i.
Fie functia SUM(V) = $\sum_{(x,y) \in V} x$, unde V este o multime de obiecte - perechi (cost, valoare)

Fie multimea $A = \{(c_1^1, c_1^2), (c_2^1, c_2^2), \ldots, (c_n^1, c_n^2)\}$, multimea care contine cele n perechi de costuri.

Vom demonstra corectitudinea urmatorului algoritm greedy:

Din multimea initiala A, vom forma o multime de obiecte B in
modul urmator:

- Dintr-un set de doua obiecte (c_i^1, c_i^2), daca $c_i^1 > \frac{c_i^2}{2}$ atunci vom adauga in multimea B obiectul $(c_i^2, 2)$.
- Dintr-un set de doua obiecte (c_i^1, c_i^2), daca $c_i^1 \le \frac{c_i^2}{2}$ atunci vom adauga in multimea B obiectul $(c_i^1, 1)$ si $(c_i^2 - c_i^1, 1)$.

Vom ordona elementele multimii B in modul urmator: sortam obiectele dupa cost / valoare.

Fie $O = \{ (c_{k_1}^{x_1}, x_1), (c_{k_2}^{x_2}, x_2), (c_{k_3}^{x_3}, x_3), \ldots, (c_{k_y}^{x_y}, x_y) \}$ solutia data de algoritmul greedy precizat anterior.

Stim ca $SUM(O) = c$ sau $c + 1$ (costul initial al functiei)
Daca SUM(O) = c + 1 atunci putem alege un obiect de valoare 2

pp. prin absurd ca exista o solutie mai buna.

Fie
$X = \{(c_{k_j}^{x_j}, x_j) \mid 1 <= j <= y\}$
$Y = \{(c_k^x, x) \mid 1 \le k \le n, k \notin \{k_j \mid 1 \le j \le y\}, 1 \le x \le 2\}$
a.i. SUM(X) = SUM (Y)

Fie P o solutie mai buna -> $P = O - X + Y$, $SUM(P) = SUM(O) - SUM(X) + SUM(Y) = SUM(O)$

Pentru ca SUM (X) = SUM(Y) rezulta ca:

I) Exista un obiect de greutate 2, $(c_{k_j}^2, 2) \in X$ a.i. $c_{k_j}^2 > c_{y_1}^1 + c_{y_2}^1$, $(c_{y_1}^1, 1), (c_{y_1}^1, 1) \in Y$
Fie $c_{y_1}^1 < c_{y_2}^1$.
Avem ca $c_{y_1}^1 + c_{y_1}^1 + (c_{y_2}^1 - c_{y_1}^1) < c_{k_j}^2 \Rightarrow$

$$c_{y_1}^1 * 2 + (c_{y_2}^1 - c_{y_1}^1) < c_{k_j}^2 * 2 \Rightarrow c_{y_1}^1 + \frac{c_{y_2}^1 - c_{y_1}^1}{2} < \frac{c_{k_j}^2}{2} \Rightarrow c_{y_1}^1 < \frac{c_{k_j}^2}{2}$$

Dar, din relatia de selectie a obiectelor, daca $c_{y_1}^1 < \frac{c_{k_j}^2}{2}$ si ele fac parte din acelasi set, atunci se vor adauga doua obiecte de greutate 1, deci aceste doua obiecte nu fac parte din acelasi set. Din selectia primilor termeni din greedy, se observa facptul ca $(c_{y_1}^1, 1)$ ar fi fost selectat, deci nu poate face parte din multimea Y.

Deci pentru acest caz nu exista o solutie mai buna.

I) Exista doua obiecte de greutate 1, $(c_{k_1}^1, 1), (c_{k_2}^1, 1) \in X$a. i. $c_{k_1}^1 + c_{k_2}^1 > c_y^2$
Fie$c_{k_1}^1 < c_{k_2}^1$
Avem ca $c_{k_1}^1 + c_{k_1}^1 + (c_{k_2}^1 - c_{k_1}^1) > c_y^2 \Rightarrow c_{k_1}^1 + \frac{(c_{k_2}^1 - c_{k_1}^1)}{2} > \frac{c_y^2}{2}$.
Dar, din relatia de selectie a obiectelor, daca $c_{k_1}^1 > \frac{c_y^2}{2}$ si ele fac parte din acelasi set, atunci se vor adauga un obiect de greutate 2, deci obiectele nu fac parte din acelasi set. Din selectia primilor termeni din greedy, se observa facptul ca $(c_{y_1}^1, 2)$ar fi fost selectat, deci nu poate face parte din multimea Y.