Documentație-Tema Greedy-Problema 9

Pentru a găsi expresia de valoare maximă am sortat inițial ambele mulțimi descrescător. Astfel, parcurgand element cu element ambele mulțimi simultan vom obține produse de valoare maximă pe care le putem adăuga în sumă. După sortare am reținut indicii primei apariții a unui element negativ în cei doi vectori sortați. De asemenea am utilizat două variabile de maxim care reprezintă numărul de elemente din cei doi vectori care vor ajuta calcularea expresiei în viitor. Am inițializat și o variabilă „toate” cu valoare 1 care are ca scop să comunice căre final dacă cei doi vecori conțin unun doar elemente pozitive iar celălalt doar elemente negative. Pornind de la începutul vectorilor parcurgem element cu element, iar dacă ambele elemente sunt pozitive putem adăuga produsul lor în sumă și să transformăm variabila „toate” in 0 pentru că avem măcar un element de același semn în ambii vectori. Parcurgem acum dinspre final vectorii element cu element, iar dacă ambele elemente sunt negative putem adăuga produsul lor în sumă și să transformăm variabila „toate” in 0 pentru că avem măcar un element de același semn în abii vectori. La final, dacă variabila „toate” a rămas 1 în funcție de care vector conține elementele negative, expresia va lua valoarea egală cu produsul dintre cel mai mic număr din vectorul cu elemente pozitive care se află la finalul vectorului sortat descrescător și cel mai mare număr din vectorul cu elemente negative care se află la începutul vectorului sortat descrescător.

Consider că soluția aleasă este optimă pentru că adăugarea unui produs în sumă se face parcurgând simultan cei doi vectori și verifiând dacă elementele respectă criterii de semn. În acest mod complexitatea nu depășește complexitatea sortării descrescătoare folosind funcția qsort. De asemenea, varianta în care nu sortăm vecorii se poate rezolva doar dacă la fiecare element al unui vector am parurge celălalt vector în întregime pentru a găsi elemente potrivite și astfel complexitatea s-ar mări.

Întrucât programul utilizat parcurge vectorii simultan obținem pentru parcurgere o complexitate O(n) și astfel complexitatea programului este dată de complexitatea operației de sortare a funcției qsort care este O(nlogn).

Pentru exemple avem mai multe cazuri pe care le exemplific mai jos:

|  |  |
| --- | --- |
| date.in | date.out |
| 5 3  5 -3 8 -1 2  2 4 3 | 51 |
| 5 3  -5 -3 -8 -1 -2  2 4 3 | -2 |
| 5 3  -5 -3 -8 -1 -2  -2 -4 -3 | 53 |
| 5 3  5 3 8 1 2  2 4 3 | 53 |
| 5 3  5 -3 8 -1 2  2 -4 3 | 46 |