

Numerele din fisierul de intrare se proceseaza la citire, pentru fiecare verificandu-se daca este putere de 3 sau de 5. Verificarea unui numar se poate face prin impartiri repetate la 3 respectiv 5, pe testul cel mai defavorabil facandu-se $\log_3(2 \text{ miliarde})$ împartiri (maxim 19). Tinand cont de numarul mare de valori de testat, se poate face urmatoarea optimizare: se memoreaza ordonate crescator toate puterile de 3 si de 5 mai mici decat 2 miliarde (sunt in total 32 valori), astfel verificarea unui numar putandu-se realiza prin cautare binara in acest sir (maxim 5 incercari).

Pentru fiecare număr putere de 3 sau de 5 vom memora într-un vector valorile 3 respectiv 5, astfel fiind necesari maxim 40000 octeti.

Avand construit acest sir, numararea secventelor care indeplinesc conditiile problemei se poate realiza in mai multe feluri, cu nivele de eficienta diferite. Dacă notăm cu m numărul de elemente din șirul nou format și cu x cel mai mic număr natural cu proprietatea că $2^x > m$, avem:

- a) Solutie cu complexitatea $m \cdot x$: calculam numarul de secvente care indeplinesc proprietatile, mai intai pentru lungimea 2, apoi 4, 8 16 ... Pentru o lungime L , procedam astfel: calculam numarul de elemente 3 (ct3) respectiv 5 (ct5) din prima secventa cu aceasta lungime. Pentru celelalte secvente, se actualizeaza ct3 si ct5 luind in calcul o noua valoare la capatul din dreapta al secventei si eliminand una de la capatul din stanga. Pe parcurs se numara secventele pentru care $ct3=ct5$. Această soluție obține 100% din punctaj.
- b) Solutie $m^2 \cdot x$ (brute force). Construim toate secventele ce incep cu pozitia fixată și au lungimea 2, numarand pentru fiecare aparitiile celor doua valori. Apoi dublăm lungimea și procedăm similar, cât timp lungimea e mai mică decât m . Testele au fost astfel alese încât aceeași soluție să obțină 30% din punctaj.
- c) Solutie $m \cdot x$. Dupa etapa de verificare a datelor din fisier memoram 1 pentru puterile lui 3 si -1 pentru puterile lui 5. Pentru fiecare secventa de lungime 2 calculam suma elementelor sale, adunand valorile din sir aflate la distanta 1 si memoram aceste sume la inceputul vectorului. Folosind noile sume, putem calcula pentru fiecare secventa de lungime 4 suma elementelor, adunand valori din noul sir aflate la distanta 2. Continuum procedeul, la pasul urmator calculam sume ale secventelor de lungime 8, adunand elemente din sir aflate la distanta 4 ... Pe parcurs se contorizeaza sumele cu valoarea 0. Pantru teste in care ponderea uneia dintre valori este mare, memorarea elementelor din sir necesita mai mult de un octet. Numarul maxim de valori fiind 40000, solutia prezentata nu se incadreaza pentru toate testele in limitele a 64k, fiind necesara utilizarea si a altor zone de memorie.

Marius Nicoli