Descrierea Soluțiilor
Olimpiada Societății pentru Excelență și Performanță în Informatică
Etapa barajelor pentru selecția echipelor naționale
Baraj 2 Juniori

1 Problema Pian

Propunători: Ioan-Cristian Pop, Universitatea Politehnica din București
Theodor-Gabriel Tulbă-Lecu, Universitatea Politehnica din București

Cerința 1 – 19 puncte

Pentru a rezolva cerința 1, trebuie să facem următoarea observație:

$$cmmdc(a, b, c) = cmmdc(cmmdc(a, b), c) \implies cmmdc(a, b) \geq cmmdc(a, b, c)$$

Astfel, valoarea minimă pe care o putem obține în șir este $g = cmmdc(t[1], t[2], \ldots, t[N])$. De asemenea, odată format acest număr g în șir putem să extindem acest număr pe întregul sir, întrucât $cmmdc(t[i], g) = g,\ \forall\ i \in \{1, 2, \ldots, N\}$.

În final, răspunsul pentru cerința 1 este $N \cdot cmmdc(t[1], t[2], \ldots, t[N])$.

Cum funcția $cmmdc(a, b)$ are complexitate $\mathcal{O}(log(min(a, b)))$, complexitatea temporală este: $\mathcal{O}(N \cdot log(MaxVal))$

1.1 Cerința 2

Pentru a rezolva cerința 2, ne vom folosi de observațiile făcute la cerința 1.

Dacă o subsecvență $(i, j), i < j$, de lungime L are proprietatea că $cmmdc(t[i], t[i+1], \ldots, t[j]) = g$, atunci această subsecvență poate crea 2 g-uri adiacente pe pozițiile $j - 1$ și j în $L - 1$ operații, aplicând *cmmdc* între două elemente consecutive de la stânga la dreapta. Odată realizat acest lucru, putem transforma întreg șirul in g în $N - 2$ operații, în total fiind necesare $N + L - 3$ operații.

Astfel, problema se reduce la a găsi lungimea minimă a unei subsecvențe care are proprietatea descrisă mai sus.

Subtask 1 ($N \leq 1\ 000$) – 38 de puncte

Putem să trecem prin toate subsecvențele din șir și să verificăm dacă este o subsecventă valid, iar pe parcurs reținem subsecvența validă de lungime minimă.

Subtask 2 ($N \leq 100\ 000$) – 42 de puncte

Ambele soluții oficiale folosesc următoarea observație:

Dacă există o o secvență de lungime L, ce respectă proprietatea că *cmmdc*-ul pe subsecvența este g, atunci va exista si o subsecvența de lungime $> L$. Astfel, putem căuta binar L-ul minim.

Soluția 1

Putem face următoarea observație: Exceptând factorii primi ai lui g, pentru ca o subsecvență să aibă *cmmdc*-ul egal cu g, niciun alt număr prim nu va apărea în factorizarea tuturor numerelor din acea subsecvență.

Acest lucru poate fi implementat utilizând un vector de frecvență pentru factorii primi care apar în factorizarea numerelor din subsecvență, condiția fiind: $fr[p] < L$, unde L este lungimea secvenței.

În plus, dacă fixăm o lungime L, și am calculat vectorul de frecvență pentru intervalul (i, j), putem trece la intervalul $(i+1, j+1)$, adăugând la vectorul de frecvență factorizarea lui $t[j+1]$, si eliminând din vectorul de frecvență factorizarea lui $t[i]$, deci putem crea un *sliding window*, pentru a calcula in $\mathcal{O}(N)$ dacă există o secvență de lungime L validă.

Precalcularea factorizărilor se poate face utilizând un algoritm asemănător cu ciurul lui Eratostene:

$$ciur[i][j] = \texttt{al j-lea număr prim care apare în factorizarea lui i}$$

$$cnt[i] = \texttt{numărul de factori primi distincți din factorizarea lui i}$$

```
for(int i = 2; i <= MaxVal; i++) {
  if(cnt[i] == 0) {
    cnt[i] = 1;
    ciur[i][1] = i;
    for(int j = 2 * i; j <= MaxVal; j += i) {
      cnt[j]++;
      ciur[j][cnt[j]] = i;
    }
  }
}
```

Putem determina și în complexitate $\mathcal{O}(N \cdot \sqrt{MaxVal})$ divizorii primi ai fiecărui numar din șir.

Soluția 2

Putem construi un *Range Minimum Query* astfel încât:

$$RMQ[i][j] = cmmdc(t[j], t[j+1], \ldots, t[j+2^i-1])$$

. Putem face acest lucru deoarce funcția *cmmdc* este o funcție idempotentă, adică $cmmdc(cmmdc(a,b),b) = cmmdc(a,b)$, proprietate necesară RMQ-ului.

Astfel, vom putea să căutăm binar subsecvența de lungime minimă si să verificăm rapid la fiecare pas utilizând RMQ-ul, dacă pentru o anumită lungime există o subsecvență care are *cmmdc*-ul g.

Atat complexitatea temporală cât și cea spațială sunt $\mathcal{O}(N \cdot log(N))$

Caz particular – $L \leq 2$

Caz particular: există numarul g în șir și/sau există doua numere consecutive care au *cmmdc*-ul egal cu g. În acest caz problema se poate rezolva în mai puțin de $N + L - 3$ instrucțiuni.

Determinăm perechile de numere care au *cmmdc*-ul egal cu g. Plecând de la stânga la dreapta: dacă există două numere consecutive care au *cmmdc*-ul egal cu g, iar ambele sunt diferite de g, atunci le transformăm pe ambele in g. Apoi, de la fiecare g din șir ne extindem la stânga și/sau la dreapta, transformând pe rând perechi de numere consecutive de forma $(*, g)$ sau $(g, *)$ in (g, g), $*$ fiind un termen oarecare din șir diferit de g.

2 Problema SP

Propunător: prof. Marius Nicoli, Colegiul Național "Frații Buzești", Craiova

Începem printr-o observație pe care o vom folosi la rezolvare și la explicațiile ulterioare din acest material: dacă o secvență [*i*,*j*] (formată din elementele cu indici de la i la j inclusiv) este palindromică, atunci toate secvențele mai scurte care sunt centrate în același loc cu ea sunt palindromice.

Facem o a doua observație: dacă secvența [*i*,*j*] devine palindromică prin extindere, este suficient să o extindem doar la unul dintre capete. De exemplu, dacă notăm cu S caracterele din secvența neextinsă, cu L caracterele folosite pentru extindere în stânga și cu R caracterele folosite la extinderea în dreapta, și dacă secvența $L_1\ L_2\ L_3\ L_4\ S_1\ S_2\ S_3\ S_4\ R_1\ R_2$ prin extindere este palindromică, atunci este necesar să avem $L_1 = R_2$ și $L_2 = R_1$, deci și secvența $L_3\ L_4\ S_1\ S_2\ S_3\ S_4$ este palindromică. Deci este suficientă extindere doar într-o parte. Valoarea K fiind mică, această observație nu era neapărat necesară pentru obținerea punctajului maxim.

Subtask 1 & 2 ($N, M \leq 2\ 000$) – 34 de puncte

O primă abordare este să verificăm pentru fiecare interogare, în timp liniar, dacă aceasta este palindromică. Complexitatea temporală fiind: $\mathcal{O}(M \cdot N \cdot K)$.

Subtask 3 & 4

A doua abordare este să precalculăm pentru fiecare poziție i:

- Lungimea maximă a unui palindrom care este centrat în i (deci de lungime impară)
- Lungimea maximă a unui palindrom centrat între pozițiile i și $i+1$ (deci de lungime pară)

Odată realizat acest lucru, pentru a testa dacă [*i*,*j*] este secvență palindromică, vom verifica dacă [*i*,*j*] este inclus în [$LeftMid$, $RightMid$] unde $LeftMid$ și $RightMid$ sunt capetele palindromului maxim care are același centru cu al secvenței [*i*,*j*] și în plus $LeftMid - i = j - RightMid$. Având K mic, acest test îl putem face pentru fiecare variantă de extensie cu maxim K caractere.

Precalcularea se poate realiza astfel:

Subtask 3 ($N \leq 2\ 000$, $M \leq 200\ 000$) – 45 de puncte

Pentru fiecare poziție i considerată ca și centru (o dată pentru palindrom de lungime impară și o dată alături de $i+1$ pentru palindrom de lungime pară) ne extindem cât mai mult în stânga și în dreapta cât timp avem caractere egale.

Timpul pentru precalculare este $\mathcal{O}(N^2)$ deci în total avem complexitatea: $\mathcal{O}(N^2 + M \cdot K)$.

Subtask 4 ($N, M \leq 200\ 000$) – 21 de puncte

Pentru o precalculare în $\mathcal{O}(N)$ putem folosi algoritmul lui Manacher. Complexitatea finală de calcul este de ordin $\mathcal{O}(N + M \cdot K)$.

O altă soluție alternativă este să calculăm coduri hash pentru secvențe din șirul dat, în ambele sensuri de la poziția 1 la poziția i și de la poziția N la poziția i și putem răspunde la fiecare întrebare deasemenea în timp constant. Această soluție permite obținerea punctajului maxim, complexitatea temporală fiind: $\mathcal{O}(N + M \cdot K)$.

3 Problema Fete și Băieți

Propunător: prof. Mihai Bunget, Colegiul Național "Tudor Vladimirescu", Târgu-Jiu

După multiplicarea de N ori a şirului s se obţine un şir format din $N \cdot K$ caractere. Cum N este divizibil cu D, există un număr natural nenul M astfel încât $N = D \cdot M$. Atunci vom avea un şir format din $N \cdot K = M \cdot K \cdot D$ caractere, care aşezate pe linii de câte D caractere va conduce la formarea unei matrice cu $M \cdot K$ linii şi D coloane. Deoarece pe primele K linii avem în total $K \cdot D$ caractere, deducem că şirul s se va repeta de exact D ori pe primele K linii. Vom numi aceasta matrice de bază. Se observă că matricea de bază se repetă de exact M ori pentru a conţine toate caracterele şirului multiplicat.

Subtask-ul 1 ($K = D$) – 23 de puncte

Cum lungimea K a şirului de caractere s este egală cu numărul de coloane al matricei, deducem că secvenţele de caractere egale cu $'b'$ de pe prima linie se suprapun cu cele de pe liniile următoare, pentru fiecare astfel de secvenţă formându-se câte o *gaşcă*. Deci e suficient să numărăm secvenţele de caractere $'b'$ din şirul s.

Complexitate $\mathcal{O}(K)$.

Subtask-ul 2 ($N = D$ şi $K < D$) – 31 de puncte

Cum $N = D$ deducem că şirul multiplicat completează doar matricea de bază. Pentru aflarea numărului de găşti se poate aplica algoritmul *fill*.

Complexitate $\mathcal{O}(K \cdot D)$.

Subtask-ul 3 (fără alte restricţii) – 46 de puncte

Se aplică algoritmul *fill* pentru matricea de bază, numărul de găşti obţinut multiplicându-l cu M. Din acest număr trebuie scăzut numărul de găşti care se pierd la alipirea a două matrice consecutive, număr multiplicat cu $M-1$ (numărul de alipiri a două matrice consecutive). Pentru a afla numărul de găşti care se pierd la alipirea a două matrice consecutive, se va forma o matrice cu $2 \cdot K$ linii, formată din două matrice de bază. Se parcurge ultima linie din prima matrice în paralel cu prima linie din a doua matrice, iar pentru fiecare băiat care se află într-o anumită *gaşcă*, aflată anterior cu algoritmul *fill*, se aplică din nou algoritmul *fill* care umple *găştile* din cele două matrice care se lipesc la frontieră (marcând astfel băieţii din aceste *găşti*). Se continuă algoritmul pentru toţi băieţii de pe frontieră care încă nu au fost marcaţi.

Ca soluţie alternativă pentru a număra câte *găşti* dispar la alipirea a două matrice se poate face *fill* pe două matrice lipite, obţinând numărul de *găşti*. Acest număr îl scădem din dublul numărului de *găşti* dintr-o matrice de bază.

Complexitate $\mathcal{O}(K \cdot D)$.

Echipa

Setul de probleme pentru această rundă a fost pregătit de:

- prof. Boian Dumitru Flavius, Colegiul Național "Spiru Haret", Târgu-Jiu
- prof. Bunget Mihai, Colegiul Naţional "Tudor Vladimirescu", Târgu-Jiu
- prof. Chește Ciprian, Liceul Tehnologic "Grigore C. Moisil", Buzău
- prof. Costineanu Veronica Raluca, Colegiul Național "Ștefan cel Mare", Suceava
- prof. Dumitrascu Dan Octavian, Colegiul National Dinicu Golescu, Câmpulung
- prof. Iordaiche Cristina, Liceul Teoretic " Grigore Moisil" Timisoara, Timișoara
- prof. Lica Daniela, Centrul Județean de Excelență, Prahova
- prof. Nicoli Marius, Colegiul Național "Frații Buzești", Craiova
- prof. Nodea Gheorghe-Eugen, Colegiul Național "Tudor Vladimirescu", Târgu-Jiu
- prof. Pițˇ-Rada Ionel-Vasile, Colegiul Național "Traian", Drobeta-Turnu Severin
- student Pop Ioan-Cristian, Universitatea Politehnica București, Facultatea de automatică și calculatoare.
- prof. Pracsiu Dan, Liceul Teoretic Emil Racoviță, Vaslui
- student Tulbă-Lecu Theodor-Gabriel, Universitatea Politehnica București, Facultatea de automatică și calculatoare.