

Solutie Nakhla

Problema se reduce la jocul NIM jucat pe N gramezi, fiecare gramada fiind formata dintr-un numar de monede egal cu spatiul liber dintre jetonul alb si jetonul negru de pe fiecare rand. Dupa cum se stie, exista strategie de castig pentru o configuratie daca si numai daca suma xor a gramezilor este diferita de 0. Fie X suma xor a celor N gramezi la un moment dat. Daca X este diferit de 0, atunci trebuie sa gasim o gramada $G[i]$ astfel incat $X \text{ xor } G[i] < G[i]$ pentru a efectua o mutare care aduce suma xor la 0. Pentru a se respecta conditia mentionata trebuie ca $G[i]$ sa aiba setat bitul cel mai semnificativ de 1 din numarul X .

Datorita numarului mare de gramezi aceasta gramda trebuie gasita intr-o complexitate $O(\lg N)$. O solutie cu $O(N)$ memorie este urmatoarea:

- se mentine un arbore de intervale care retine in fiecare nod xor -ul pe acel interval
- pentru a gasi o gramada care are setat cel mai semnificativ bit din X se face o parcurgere in arbore, la fiecare pas mergand pe subarborele care are setat acel bit (nu-l pot avea setat ambii subarborii)
- dupa ce se face o mutare se actualizeaza arborele de la frunza care reprezinta gramada respectiva pana la radacina

O alta solutie cu $O(\lg N)$ pe mutare, dar cu $O(N \lg N)$ memorie este urmatoarea:

- se pastreaza cate o lista pentru fiecare bit de la 0 la 30 cu indicii gramezilor care contin acel bit
- pentru a efectua o mutare trebuie sa ne uitam in lista corespunzatoare bitului cel mai semnificativ al lui X
- dupa ce se face o mutare se sterge elementul din listele in care era inainte si se insereaza in alte liste, conform noii valori
- pentru a implementa usor acest algoritm se poate folosi o tehnica de "lazy deletion": dupa o mutare nu se sterg elementele din lista; atunci cand vrem sa efectuam o mutare scoatem dintr-o lista pana cand gasim un element care chiar contine acel bit