**Descriere soluție – margi**

*Propunător:*

*prof. Eugen Nodea*

*Colegiul Național “Tudor Vladimirescu”, Târgu Jiu*

Metoda de rezolvare este ”Divide et impera”.

Vom numi **drum** = secvența de submatrici alese pentru a ajunge de la nivelul n la nivelul **1**.

Se observă că soluția finală poate fi compusa în două moduri:

1. Cele doua drumuri sunt complet disjuncte, adică la nivelul n jucătorul **A** si jucătorul **B** aleg submatrici diferite, iar in continuare este evident ca drumurile lor nu se vor mai intersecta
2. Cei doi aleg aceiași submatrice câteva nivele, după care la un nivel **X<n**, drumurile lor se ”despart” și cei doi jucători aleg submatrici diferite

Așadar, funcția noastră trebuie sa rezolve aceste doua cazuri simultan, și să aleagă întotdeauna soluția ”minima lexicografic”. Aceasta soluție se obține dacă se merge pe principiul ”primul venit, primul servit”, adică se apelează întotdeauna funcția de divide pe submatrici in aceasta ordine **1, 2,3,4.**

Funcția recursiva va avea **4** parametrii (**x, y, niv, sumaAnt**) adică coordonatele coltului stânga sus al submatricei curente sunt **x y**, suntem la nivelul **niv** si, *cel mai important*, **sumaAnt** suma care se obține daca ambii jucători ajung in aceasta submatrice la nivelul **niv** (deci amândoi au ales aceleași submatrici pana la nivelul **niv** si suma lor va fi egala cu suma submatricilor alese pana la acest nivel).

Acum, **suma\_maxima=max(suma\_maxima, sumaAnt + cele mai mari doua sume obținute in doua submatrici din cele 4** in care se împarte submatricea actuală cu coltul in **x y**). Funcția returnează suma maxima care poate fi obținuta daca un jucător începe drumul la acest nivel.

De fapt, prin apelurile recursive ale funcției divide obținem un arbore.

**N=2^n**

Complexitate: **(N)\* log (N)**

Memorie: **(2^n)\*(2^n)**