* Descrierea programului:

Abordare Divide et Impera pentru rezolvarea problemei 3, var. 3;

Complexitate – *O(n log n)*, unde *n* este lungimea secvenței de intrare.

* Descrierea funcției recursive:

Se observă posibilitatea numărării inversiunilor în timp liniar într-o permutare „semiordonată” (compusă din două părți ordonate crescător);

Considerăm următorul vector „semiordonat”:

*x = , și ;*

Pentru a obține numărul de inversiuni trebuie interclasate cele două jumătăți ale vectorului în urma căreia se obține un vector ordonat crescător (timp liniar);

Funcția de interclasare se poate rezuma la un singur pas general aplicabil de mai multe ori:

Se adaugă un element din jumătatea dreaptă a vectorului, pe urmă se adaugă *i* elemente din jumătatea stângă a vectorului:

*y y* ∔, unde *y* este un vector ordonat obținut în pașii anteriori, iar ∔ este operatorul de concatenare;

Fiindcă este mai mare decât toate elementele din stânga sa (in vectorul „semiordonat” *x*) și elementele sunt mai mari decât (în vectorul ordonat *y*) și, implicit, decât , numărul de inversiuni obținute la acest pas este .

Însă, pentru a calcula permutările semnificative ale vectorului *x* trebuie „simulată” interclasarea următorului vector auxiliar obținut din *x* :

*x’ =*

*S*e observă că, dacă la oricare pas al interclasării, un vector ordonat *y’* are următoarea formă:

*y’ y’* ∔

Atunci la pasul respectiv s-au găsit inversiuni semnificative.

În acest program, funcția recursivă *divide(int[])* primește ca parametru un vector pe care-l împarte în două jumătăți până când obține doi vectori triviali ordonați (cu un singur element) pentru care apelează funcția *merge(int[],int[])* care numără inversiunile semnificative din concatenarea celor doi vectori și apoi interclasează cei doi vectori presupuși ordonați crescător (de la pașii anteriori).

* Demonstrația complexității:

Folosind Teorema Master: *T(n) = aT(n/b) + f(n)*

În particular, pentru acest algoritm:

*a = 2* : se apelează funcția *divide(int[])* în ea însăși de 2 ori: pentru prima, respectiv a doua jumătate a vectorului parametru;

*b = 2* : lungimea vectorului parametru se împarte la 2 pentru fiecare reapelare a funcției *divide(int[])*;

*f(n) = Θ(n)*: funcția *merge(int[],int[])* apelată în *divide(int[])* are complexitate *O(n).*

Rezultă din Teorema Master *T(n) = O(n log n)*.