* Descrierea programului:

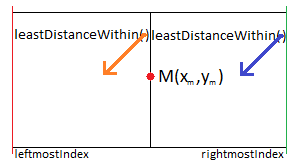
Abordare Divide et Impera pentru rezolvarea problemei 4, var. 3;

Complexitate – *O(n log n)*, unde *n* este numărul de puncte date la intrare.

* Descrierea funcției recursive:

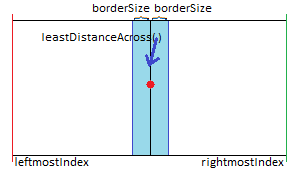
Considerăm o hartă M generată de punctele date ca intrare; funcția recursivă divide harta în subcategorii delimitate de mediană și primește ca parametri punctele din harta curentă, indexul cel mai din stânga și, respectiv, cel mai din dreapta al hărții de divizat.

* Cazul de continuare:



Pentru indecșii primiți ca parametri, funcția calculează cea mai mică distanță pentru subcategoria dintre cel mai din stânga index și mediană, respectiv, dintre mediană și cel mai din dreapta index;

Pentru tratarea cazului în care distanța dintre cele mai apropiate 2 puncte trece „peste graniță” se folosește funcția *leastDistanceAcross(Coordinate[],double,int,int)* ce calculează în timp liniar distanța minimă dintre două puncte aflate pe „graniță”; Se păstrează complexitatea liniară în timp datorită faptului că pentru fiecare punct de pe „graniță” se calculează distanța față de următoarele 7 în ordine crescătoare după coordonata *y*; De asemenea, funcția primește ca parametru punctele din harta curentă sortate după *y* și, astfel, „filtrarea” punctelor din „graniță” se face tot în timp liniar.



În final, funcția returnează minimul dintre cele trei distanțe obținute.

* Cazul de oprire:

Atunci când într-o subcategorie se află doar 2 puncte (cele de pe indecșii marginali) se returnează efectiv distanța între acestea.

* Demonstrația complexității:

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

În particular, pentru acest algoritm:

*a = 2* : se apelează funcția *leastDistanceWithin(int,int)* de două ori în ea însăși: pentru subcategoria dinaintea medianei, respectiv subcategoria de după mediană;

*b = 2* : harta devine de două ori mai restrânsă cu fiecare reapelare a funcției *leastDistanceWithin(int,int)*;

*f(n) = Θ(n)* : în funcția *leastDistanceWithin(int,int)* se mai apeleazăfuncția *leastDistanceAcross(doube,int,int)*: *O(n)*;

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