|  |  |
| --- | --- |
| 4. Пусть дана функция  int F(int n,int x);  {    int i, j, k, m;                 // 4    k = m = 0;                      // 2    int i,j,k,m,n,s=0;              // 7      for (i = 1; i <= 2\*n; i++)      // 1 + sum(1\_2\*N)(1+1+...)    {      k++;                            // 1      if (a[i] > х)                   // 2      {        for (j = 1; j <= n\*n; j++)    // 1 + sum(1\_N\*N)(1+1+...)        {          s = s + a[j];                 // 3          m++                           // 1        }      }    }    return k+m;                      // 2  }  Определите функцию роста *f*(*N*) трудоемкости данного алгоритма и её асимптотические оценки Θ( *f*(*N*)), O(*f*(*N*)), Ω( *f*(*N*)), o(*f*(*N*)), ω( *f*(*N*)), где N – длина входа. | Решение:  *// мы не знаем, чему равно a[i], так что более точную оценку дать сложно*  *f*(*N*)=  Θ(*f*(*N*)) = O(*f*(*N*)) = Ω( *f*(*N*)) = n3  o(*f*(*N*)) = n2, ω( *f*(*N*)) = |
| 5. Вычислите функцию роста *f*(*N*) трудоемкости данного алгоритма с учетом *фактического* числа инструкций, исполняемых при выполнении функции, анализ трудоемкости и расчет *f*(*n*) которой был Вами выполнен в задании №4  int t,u;                       // 2  for (q = 1;q <= 10; q++)       // 1 + sum(q=1\_10)(1+1+...)  {    F(5, q);              // 8n^3 + 12n + 16 = 1000 + 60 + 16 = 1076    for (t = 1; t <= 20; t++)      // 1 + sum(t=1\_20)(1+1+...)    {      F(q, 2\*t);                  //  n^3 + 12n + 16    }  } | Решение:  *// мы не знаем, чему равно a[i], так что более точную оценку дать сложно*  *f*(*N*) =  Θ(*f*(*N*)) = O(*f*(*N*)) = Ω(*f*(*N*)) = 1 |