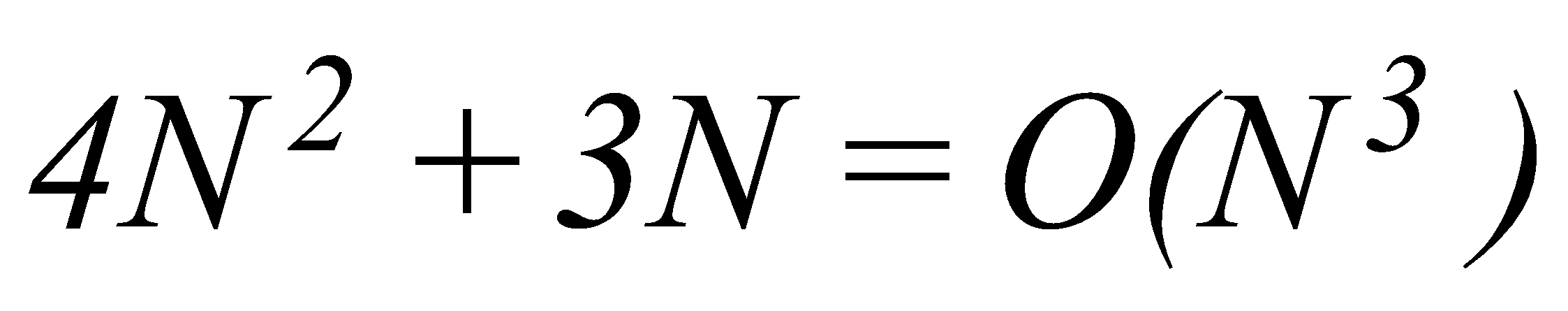
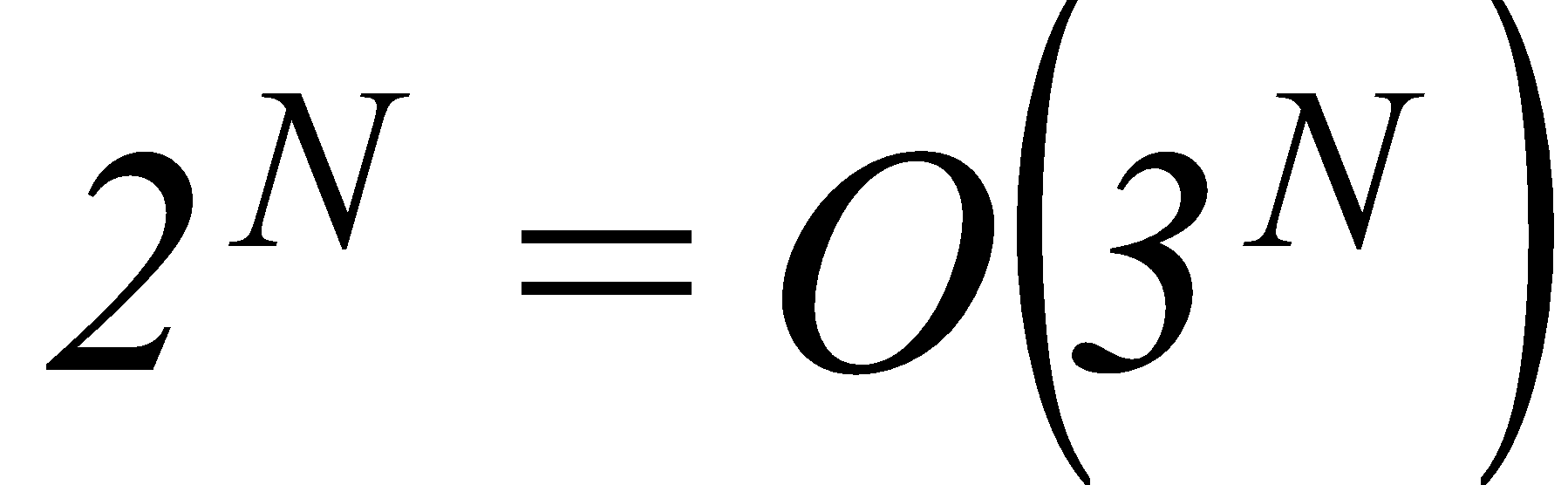
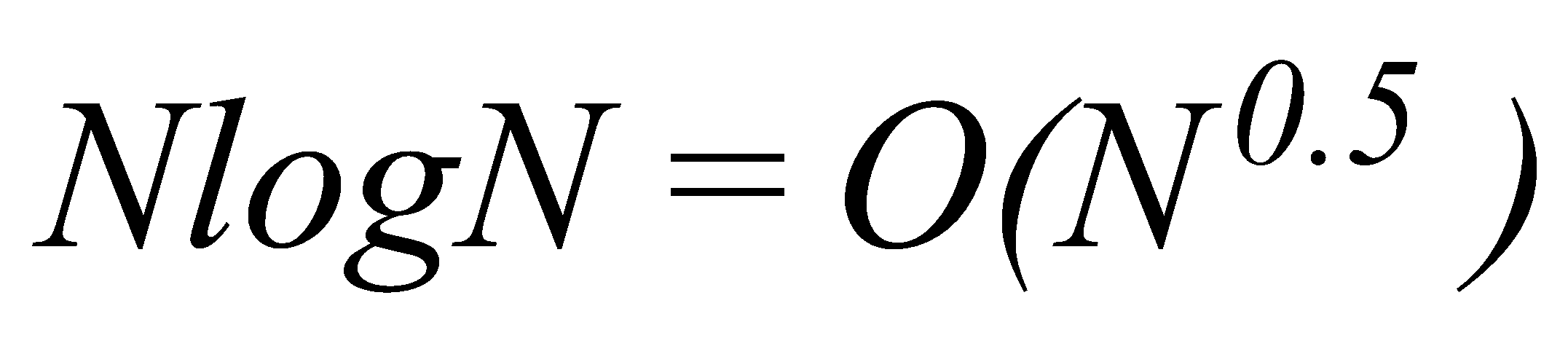
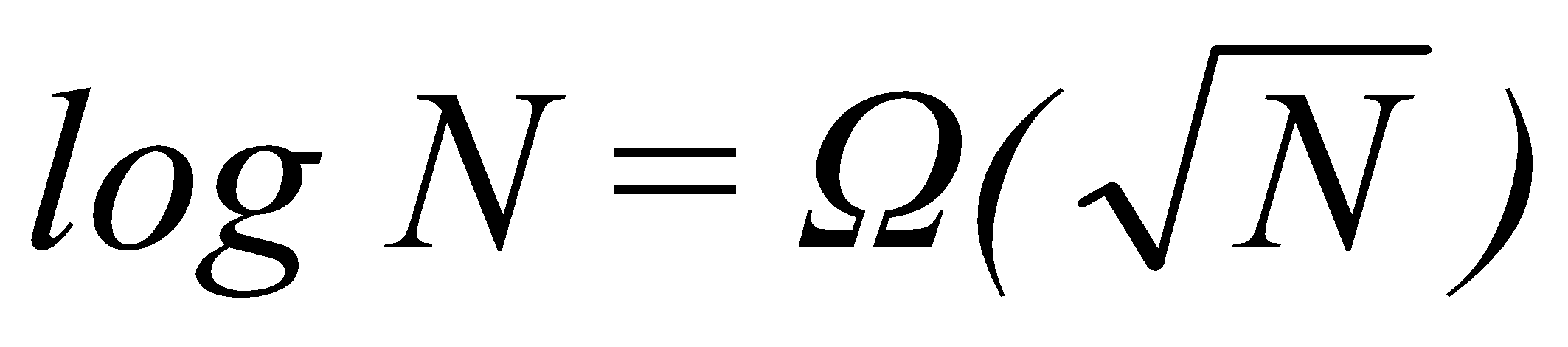
**Лабораторная работа №3**

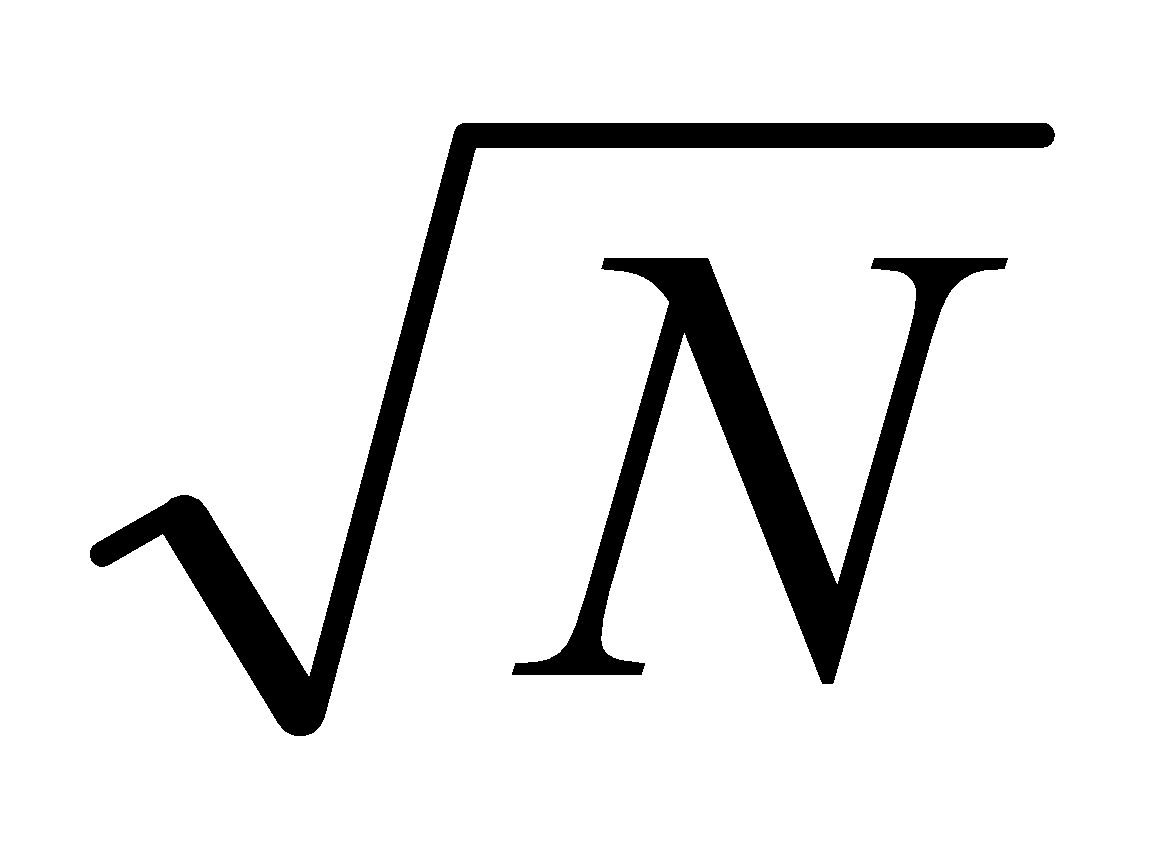
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ варианта** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **№ задания** | a,b,c,p | d,e,f,q | g,h,i,r | j,k, l,s | p,m,n,o | a,d,g,q | b,f,h,r | k,n,e,s | c,f,g,p | i,l, n,q | a,h,o,r | f,i, l,s | b,k,m,p | c,k,o,q | b,i,m,r |

1. Проверить с помощью определения:
   1. *17 = O(1)*;
   2. *N(N — 1)/2 = O(N2)*;
   3. *22N = O(2N)*;
   4. *logN = O(N)*;
   5. *N logN = O(N2)*;
   6. *N ln N = O(N3/2)*;
   7. *3N3 + 2N2 = O(N2)*;
   8. *N logN + 5 = O(N)*;
   9. *2N + N3 = O(2N)*;
   10. *f(N) = O(f(N))*;
   11. *f(N) = O(f(N)2)*;
   12. *(N + 1) log N = O(N)*;
   13.  ;
   14. ;
   15. 
   16. *N3 + 2N2 = Ω(N2)*;
   17. 
   18. *N log N = Ω(N2)*;
   19. *3N2 + 2N — 5 = Ω(N)*;
   20. *N + log N = Ω(log N)*.
2. Пусть время работы алгоритма *Т(N) = O(f(N))*. Если X элементов обрабатываются за Y мсек., то во сколько раз следует ожидать увеличения времени выполнения при обработке Z элементов?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ варианта** | **f(N)** | **X** | **Y** | **Z** | **№ варианта** | **f(N)** | **X** | **Y** | **Z** |
|  | N2 | 1000 | 5 | 3000 |  | N3 | 3000 | 50 | 9000 |
|  | N | 2000 | 10 | 5000 |  | 2N | 1000 | 100 | 2000 |
|  | N3 | 3000 | 10 | 6000 |  | logN | 5000 | 8 | 20000 |
|  | 2N | 4000 | 4 | 4002 |  | 1 | 1000 | 12 | 3000 |
|  | logN | 5000 | 2 | 15000 |  | logN | 2000 | 10 | 6000 |
|  | 1 | 1000 | 4 | 4000 |  | N3 | 1000 | 20 | 2000 |
|  | N2 | 2000 | 11 | 6000 |  | N2 | 2000 | 15 | 10000 |
|  | N | 3000 | 20 | 9000 |  | 2N | 3000 | 10 | 3003 |

1. Найти наиболее точную оценку для рекуррентных отношений.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ варианта** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **№ задания** | a,b,c,p | d,e,f,q | g,h,i,r | j,k, l,s | p,m,n,o | a,d,g,q | b,f,h,r | k,n,e,s | c,f,g,p | i,l, n,q | a,h,o,r | f,i, l,s | b,k,m,p | c,k,o,q | b,i,m,r |

* 1. *T(N) = 3T(N/2) + N, T(1) = 1*;
  2. *T(N) = 3T(N/2) + N2, T(1) = 1*;
  3. *T(N) = 8T(N/2) + N3, T(1) = 1*;
  4. *T(N) = 2T(N/2) + N logN, T(1) = 1*;
  5. *T(N) = 9T(N/2) + N3, T(1) = 1*;
  6. *Т(1) = 1, Т(N) = 2T(N/2) + 1, при N > 1*;
  7. *T(N) = 3T(N/2) + N2, T(1) = 1*;
  8. *T(N) = 3T(N/2) + N logN, Т(N) — константа при N ≤ 8*;
  9. *T(N) = 16T(N/4) + N2, если Т(N) — константа при N ≤ 2*;
  10. *T(N) = 9T(N/2) + N2, T(1) = 1*;
  11. *T(N) = 7T(N/2) + N2, если Т(N) — константа при N ≤ 2*;
  12. *T(N) = 2T(N/4) + N1/2, если Т(N) — константа при N ≤ 2*;
  13. *T(N) = 2T(N/4) +* *, T(1) = 1*;
  14. *T(N) = 2T(N/2) + N logN, если Т(N) — константа при N ≤ 2*;
  15. *T(N) = T(9N/10) + N; если Т(N) - константа при N ≤ 2*;
  16. *T(N) = 2T(N — 1) + 1, T(1) = 2*;
  17. *T(N) = T(N — 1) + N, T(1) = 1*;
  18. *T(N) = 2T(N — 1) + N, T(1) = 2*;
  19. *T(N) = T(N — 1) + log N, T(1) = 2*.

1. Используя *О-* символику, найдите время выполнения (как функции от *N*) процедуры или функции в наихудшем случае.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ варианта** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **№ задания** | a | b | c | d | a | b | c | d | a | b | c | d | b | c | d |

* 1. **procedure** matmpy (N: integer);

**var** i, j, k: integer;

**begin**

**for** i:= 1 **to** N **do**

**for** j:= 1 **to** N **do**

**begin**

C[i,j]:= 0;

**for** k:= 1 **to** N **do**

C[i,j]:= C[i,j] + A[i,k] \* B[k,j]

**end**

**end;**

* 1. **procedure** mystery (N: integer);

**var** i, j, k: integer;

**begin**

**for** i:= 1 **to** N-1 **do**

**for** j:= i+1 **to** N **do**

**for** k:= 1 **to** j **do**

*{группа операторов с временем выполнения О(1)}*

**end;**

* 1. **procedure** veryodd (N: integer);

**var** i, j, x, y: integer;

**begin**

**for** i:= 1 **to** N **do**

**if** odd(i) **then**

**begin**

**for** j:= 1 **to** N **do** x:= x+1;

**for** j:= 1 **to** i **do** y:= y+1;

**end**

**end;**

* 1. **function** recursive (N: integer): integer;

**begin**

**if** N<= 1 **then return** (1)

**else**

**return** (recursive (N-1) + recursive (N-2))

**end;**