Problem 1. (20 marks)

For each pair off(n) and g(n) below, decide iff(n) = O(g(n)) ,f(n) = Ω(g(n)) , orf(n) = Θ(g(n)) .

Justify your answer using the definitions of these asymptotic notation. Note that more than one of these relations may hold for a given pair; list all correct ones.



(2)f(n) = (log3 n)2 and g(n) = log2 (n3 ) .

(3)f(n) = 2n and g(n) = 22n .

(4)f(n) = log2 (n!) and g(n) = nlog2 n.

Problem 2. (20 marks)

Determine the Big O notation of the following code snippets:

(1)

|  |
| --- |
| void exampleFunction(std::vector<int> arr) {  for (int i = 0; i < arr.size(); i++) {  std::cout << arr[i] << std::endl;  }  for (int i = 0; i < arr.size(); i++) {  for (int j = 0; j < arr.size(); j++) {  std::cout << arr[i] << " " << arr[j] << std::endl;  }  }  } |

(2)

void fun(int N, int M) { std::vector<int> arr;

int counter = 0;

for (int i = 0; i < N; i++) {

arr.push\_back(i);

}

for (int i = 0; i < M; i++) {

counter++; }

std::cout << counter << std::endl; }

Problem 3. (20 marks)

Let f(n) and g(n) be asymptotically positive functions. Prove or disprove each of the following conjectures.

Hint: You can disprove a conjecture by giving negative examples.

1 .f(n) = O(f(n)/3) .

2.f(n) = O(f(n/3)) .

Problem 4. (20 marks)

 , where n > 1 . Problem 5. (20 marks)

Determine the Big O notation of the following recursive function in C++:

|  |
| --- |
| int fibonacci(int n) {  if (n <= 1) {  return n;  } else {  return fibonacci(n-1) + fibonacci(n-2);  }  } |