

챕터 9. Greedy Approach (탐욕 접근법) - (3) Fibonacci Decomposition

```
1 // Greedy Approach (3) Fibonacci Decomposition
2
3 #include <iostream>
4 #include <vector>
5 #include <algorithm>
6
7 using namespace std;
8
9 int main()
10 {
11     unsigned long long N;
12     cin >> N;
13
14     // 피보나치 수열 생성 (최대 값이 2^64 - 1 이하)
15     vector<unsigned long long> fib = {1, 2}; // 제켄도르프 정리에선 F(1) = 1, F(2) = 2 기준
16
17     while (true)
18     {
19         unsigned long long next = fib[fib.size() - 1] + fib[fib.size() - 2];
20         if (next > N)
21             break;
22         fib.push_back(next);
23     }
24
25     // 그리디하게 큰 수부터 선택
26     vector<unsigned long long> result;
27     for (int i = fib.size() - 1; i >= 0; --i)
28     {
29         if (fib[i] <= N)
30         {
31             result.push_back(fib[i]);
32             N -= fib[i];
33             // 연속된 피보나치 수를 피하기 위해 바로 이전 항은 skip
34             i--; // i-- 한 번 더 해서 건너뛴
35         }
36     }
37
38     // 오름차순 출력
39     sort(result.begin(), result.end());
40     for (auto num : result)
41         cout << num << '\n';
42
43     return 0;
44 }
45
```