

斐波那契编码——实验报告

- 班级：通信2301
- 学号：U202342641
- 姓名：陶宇轩

一、编程实验名称与内容概述

- 实验名称：斐波那契编码
- 内容概述：按斐波那契编码规则进行编码，从键盘输入正整数，程序运行后输出对应的斐波那契编码，以下是编码规则：
 - 将输入的正整数分解为不连续的斐波那契数之和。
 - 按斐波那契数列的顺序（从小到大）生成二进制位，每个斐波那契数对应一个位：选中的数为 1，未选中的数为 0。
 - 在二进制表示的末尾添加一个 1 作为结束符。

二、程序设计思路

数据结构

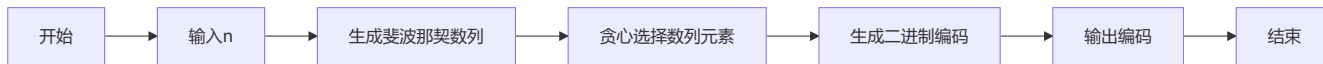
- `vector<int> fib`：用于存储斐波那契数列
- `vector<bool> selected`：标记选中的斐波那契数

算法步骤

- 生成斐波那契数列直至超过输入值 n
- 从最大斐波那契数开始，依次选择不超过剩余值的数，标记并减去该数
- 根据标记数组生成二进制字符串，末尾添加终止符 1

三、代码说明

流程图



主函数

- 输入整数 n
- 生成斐波那契数列

```

1  vector<int> fib = {1, 2};
2  while (true) {
3      int next = fib.back() + fib[fib.size()-2];
4      if (next > n) break;
5      fib.push_back(next);
6  }

```

3. 贪心选择斐波那契数

```

1  int remaining = n;
2  for (int i = max_idx; i >= 0; --i) {
3      if (fib[i] <= remaining) {
4          selected[i] = true;
5          remaining -= fib[i];
6      }
7  }

```

4. 生成编码并输出

```

1  string code;
2  for (bool used : selected) {
3      code += used ? '1' : '0';
4  }
5  code += '1'; // 添加终止符 1

```

四、运行结果与复杂度分析

运行结果

复杂度分析

- 时间复杂度：
 - 生成斐波那契数： $O(\log(n))$
 - 贪心选择斐波那契数： $O(\log(n))$
 - 总时间复杂度： $O(\log(n))$
- 空间复杂度： $O(\log(n))$ ，存储斐波那契数列和标记数组

五、改进方向与心得体会

改进方向

- 寻找复杂度更低的算法

心得体会

- 用预生成的斐波那契数列可以方便调用，节省时间