

程序员进阶之算法练习(十)--最长的斐波那契子序列的长度(暴力法)

文章出自 Hello Code 开发者学习平台 CC老师

获取最新文章/视频 关注公众号:

HelloCode开发者学习平台



一.面试题目

如果序列 X_1, X_2, \dots, X_n 满足下列条件,就说它是 斐波拉契式的:

- $n \geq 3$
- 对于所有 $i+2 \leq n$, 都有 $X_i + X_{i+1} = X_{i+2}$;

给定一个严格递增的正整数数组形成序列.找到A中最长的斐波拉契式子序列的长度.如果一个不存在,返回0.比如,子序列是从原序列A中派生出来的.它从A中删除任意数量的元素.而不改变其元素的顺序.例如 $[3, 5, 8]$ 是 $[3, 4, 5, 6, 7, 8]$ 的子序列.

二.案例

案例(1)

- 输入: [1,2,3,4,5,6,7,8]
- 输出: 5
- 原因: 最长的斐波拉契式子序列: [1,2,3,5,8]

案例(2)

- 输入: [1,3,7,11,12,14,18]
- 输出: 3
- 原因: 最长的斐波拉契式子序列: [1,11,12], [3,11,14], [7,11,18]

三.解决方案-- 使用Set(集合)暴力法

• 思路

每个斐波拉契的子序列都依靠2个相邻项来确定下一个预期项,例如,对于 2,5. 我们所期望的子序列必定以 7,12,19,31 等继续.

我们可以使用set结构来快速确定下一项是否在数组A中.由于这些项的值以指数形式增长.最大值 $\leq 10^9$ 的斐波拉契式的子序列有43项目.

• 算法

对于每个起始对 $A[i], A[j]$. 我们保持下一个预期值. $y = A[i] + A[j]$. 和此前看到的最大值 $x = A[j]$. 如果Y在数组中,我们可以更新这些值 $(x, y) \rightarrow (y, x+y)$.

注意: 由于子序列的长度大于等于3,只能是斐波拉契式的,所以我们必须进行检查 $ans \geq 3 ? ans : 0$

四.代码实现

• C++ Code

```
class Solution {
public:
    int lenLongestFibSubseq(vector<int>& A) {
        int N = A.size();
        unordered_set<int> S(A.begin(), A.end());
```

```

int ans = 0;
for (int i = 0; i < N; ++i)
    for (int j = i+1; j < N; ++j) {
        /* 使用起始a[i],a[j],
           * y代表未来预期的值
           * x代表最新的当前值
           */
        int x = A[j], y = A[i] + A[j];
        int length = 2;
        while (S.find(y) != S.end()) {
            int z = x + y;
            x = y;
            y = z;
            ans = max(ans, ++length);
        }
    }

return ans >= 3 ? ans : 0;
}
};

```

• Java Code

```

class Solution {
    public int lenLongestFibSubseq(int[] A) {
        int N = A.length;
        Set<Integer> S = new HashSet();
        for (int x: A) S.add(x);

        int ans = 0;
        for (int i = 0; i < N; ++i)
            for (int j = i+1; j < N; ++j) {
                /* 使用起始a[i],a[j],
                   * y代表未来预期的值

```

```

        * x代表最新的当前值
        */
        int x = A[j], y = A[i] + A[j];
        int length = 2;
        while (S.contains(y)) {
            // x, y -> y, x+y
            int tmp = y;
            y += x;
            x = tmp;
            ans = Math.max(ans, ++length);
        }
    }

    return ans >= 3 ? ans : 0;
}
}

```

• python Code

```

class Solution(object):
    def lenLongestFibSubseq(self, A):
        S = set(A)
        ans = 0
        for i in xrange(len(A)):
            for j in xrange(i+1, len(A)):
                x, y
                = A[j], A[i] + A[j]
                length = 2
                while y in S:
                    x, y = y, x + y
                    length += 1
                ans = max(ans, length)
        return ans if ans >= 3 else 0

```

五.复杂度分析

- 时间复杂度: $O(N^2 \log M)$,其中**N**指的是**A**的长度,**M**指的是**A**的最大值
- 空间复杂度: $O(N)$,集合S的使用空间

六.学习建议

- 理解斐波拉契式数列的规律
- 理解代码思路

只有努力不会背叛你，加油吧



