20220419-C++

1.过程描述

节点着色算法 C++ 复制代码

```cpp
//节点着色算法是根据给定的颜色种数，对图的节点进行着色的一个算法。限制条件是相连的节点的颜色不能相同



#include <array>
#include <iostream>
#include <vector>

namespace backtracking
{
    namespace graph_coloring
    {
        //V: 节点的数量
        template <size_t V> //size_t在64位操作系统中为long long unsigned int.size_t主要是为了适应多个平台
        void printSolution(const std::array<int, V>& color)
        {
            std::cout << "Following are the assigned colors\n";
            for (auto& col : color)
            {
                std::cout << col;
            }
            std::cout << "\n";
        }

        //通过在array中嵌套array得到graph
        //v:节点v
        //c:颜色值
        template<size_t V>
        bool isSafe(int v, const std::array<std::array<int, V>, V>& graph,
            const std::array<int, V>& color, int c)
        {
            for (int i = 0; i < V; i++)
            {
                if (graph[v][i] && c == color[i])
                {
                    return false;
                }
            }
            return true;
        }

        template <size_t V>
```

```
43          void graphColoring(const std::array<std::array<int, V>, V>&
    graph, int m, std::array<int, V> color, int v)
44          {
45              if (v == V)
46              {
47                  printSolution<V>(color);
48                  return;
49              }
50              for (int c = 1; c <= m; c++)
51              {
52                  if (isSafe <V>(v, graph, color, c))
53                  {
54                      color[v] = c;
55                      graphColoring<V>(graph, m, color, v + 1);
56                      color[v] = 0;
57                  }
58              }
59          }
60      }
61  }
62
63  int main()
64  {
65      const int V = 4;
66      std::array<std::array<int, V>, V> graph =
67      {
68          std::array<int,V>({0,1,1,1}),std::array<int,V>({1,0,1,0}),
69          std::array<int,V>({1,1,0,1}),std::array<int,V>({1,0,1,0})
70      };
71      int m = 3;
72      std::array<int, V> color{};
73      backtracking::graph_coloring::graphColoring<V>(graph, m, color, 0);
74      return 0;
75  }
```

Minimax C++ 复制代码

```cpp
#include <algorithm>
#include <array>
#include <cmath>
#include <iostream>
//这个算法需要看些资料好好理解
namespace backtracking
{
    template<size_t T>
    int minimax(int depth, int node_index, bool is_max, const std::array<int, T>& scores, double height)
    {
        if (depth == height)
        {
            return scores[node_index];
        }
        int v1 = minimax(depth + 1, node_index * 2, !is_max, scores, height);
        int v2 = minimax(depth + 1, node_index * 2 + 1, !is_max, scores,height);
        return is_max ? std::max(v1, v2) : std::min(v1, v2);
    }
}

int main()
{
    std::array<int, 8> scores = { 90,23,6,33,21,65,123,34423 };
    double height = log2(scores.size());
    std::cout << "Optimal value: "
        << backtracking::minimax(0, 0, true, scores, height) << std::endl;
    return 0;
}
```

2.结果输出

今天原本主要跟着github上的C++算法库写了两个算法，其中第二个极小极大值没怎么理解，感觉涉及到算法中的树之类的知识，虽然代码量很少，但可以感觉背后蕴含的思想有一定厚度。明天开始决定把邓俊辉的数据结构和算法课先看了，这两天一直在找一些练手的项目，但终究还是没能找到符合期望的，要么太复杂，要么感觉学不到太多东西。还是得提醒一下自己，CS这种东西急也急不来，基础没打牢容易四处碰壁，这也是这两天实实在在的感受。等算法跟数据结构大概摸清楚后，再来想实战的事情。