2021-10-24 作业

完全加括号的矩阵连乘积

- 完全加括号的矩阵连乘积可递归地定义为:

  （1）单个矩阵是完全加括号的;
  （2）矩阵连乘积 A 是完全加括号的，则 A 可表示为2个完全加括号的矩阵连乘积 B 和 C 的乘积并加括号，即 $A=(BC)$

- 设有四个矩阵 A, B, C, D, 它们的维数分别是:
  $A=50\times10$ $B=10\times40$ $C=40\times30$ $D=30\times5$

- 总共有五种完全加括号的方式
  $(A((BC)D))$ $(A(B(CD)))$ $((AB)(CD))$
  $(((AB)C)D)$ $((A(BC))D)$

  16000, 10500, 36000, 87500, 34500

作业:
编写程序，要求
1、计算数目;
2、输出加括号系列;
3、分析时间复杂度;

△代码如下:

```
vector<string> matrix_chain(string * p, int start, int length)
{
    vector<string> ret;
    if (length == 1) {
        ret.push_back(p[start]);
    }
    else if (length == 2) {
        ret.push_back("(" + p[start] + p[start + 1] + ")");
    }
    else {
        for (int i = 1; i != length; i++) {
            vector<string> part1 = matrix_chain(p, start, i);
            vector<string> part2 = matrix_chain(p, start + i, length - i);
            for (int k = 0; k != part1.size(); k++) {
                for (int m = 0; m != part2.size(); m++) {
                    ret.push_back("(" + part1[k] + part2[m] + ")");
                }
            }
        }
    }
    return ret;
}

//n表示矩阵链的矩阵数目
int f(int n)
{
    int num=0;
    if (n == 1)
        num = 1;
    else
        for (int i = 1; i != n; i++)
            num += f(i)*f(n - i); //左子矩阵链完全加括号的数目乘以右子矩阵链完全加括号的数目
    return num;
}

int main()
{
    int n;
    cout << "请输入矩阵个数: ";
    cin >> n;
    string* p = new string[n];  // 向量p用来存储n个矩阵
    for(int i=0; i!=n; i++)
    {
        p[i] = char(i + char('A'));
    }
    vector<string> result = matrix_chain(p, 0, n);
    cout << "理论上, 方案数量: " << f(n) << endl;
    cout << "矩阵链乘完全括号化方案数量: "<<result.size() << endl;
    for (int i = 0; i != result.size(); i++)
        cout << result[i] << endl;

    return 0;
}
```

△加括号的情况如下:

```
请输入矩阵个数: 5
理论上, 方案数量: 14
矩阵链乘完全括号化方案数量: 14
(A(B(C(DE))))
(A(B((CD)E)))
(A((BC)(DE)))
(A((B(CD))E))
(A(((BC)D)E))
((AB)(C(DE)))
((AB)((CD)E))
((A(BC))(DE))
(((AB)C)(DE))
((A(B(CD)))E)
((A((BC)D))E)
(((AB)(CD))E)
(((A(BC))D)E)
((((AB)C)D)E)
```

△ 时间复杂度分析：

设 $T(n)$，n 为矩阵的数量

由代码可知：

$$T(n) = \begin{cases} 1, & n=1 \\ \sum_{i=1}^{n-1} T(i) \cdot T(n-i), & n>1 \end{cases}$$

参考 9-12 的作业 2 的结果，可知：

$$T(n) = \frac{1}{n} C_{2n-2}^{n-1}$$

∴ 时间复杂度为 $O\left(\frac{1}{n} C_{2n-2}^{n-1}\right)$