

实验二 分治

May 2022

1 实验项目结构

- find_maximum_subarray
 - include
 - util.hpp
 - Solution.hpp
 - data
 - main.cpp
- round_robin_schedule
 - 略
- perfect_permutation
 - 略

本实验包含三个独立的题目：find_maximum_subarray（最大子序列）和 round_robin_schedule（循环赛时间表），perfect_permutation（拓展题：完美排列）。

每个题目的代码结构是类似的，include 文件夹中包含了 util.hpp 和 Solution.hpp，data 文件夹中包含了测试数据，对于每个题目，你需要完成 Solution.hpp 的编写，然后编译运行 main.cpp 来进行校验。

请注意，每次对 Solution.hpp 修改完之后，需要重新编译运行 main.cpp，如何直接执行上次编译好的 main.exe 或 main，新的修改将不会生效。

2 实验内容

2.1 最大子序列

给定一个长度为 N 的数组 A ，其任意连续子序列可以表示为 A_i, A_{i+1}, \dots, A_j ，其中 $0 \leq i \leq j \leq N - 1$ 。最大子序列是指所有的子序列中和最大的那一个，注

子序列不能为空。你需要输出它的和，即：

$$result = \max_{0 \leq i \leq j \leq N-1} \sum_{k=i}^j A[k]$$

例如：[-2, 11, -4, 13, -5, -2] 的答案为 20。

请根据课件中的伪代码思想完成 Solution.hpp 的实现。

```
class Solution {
public:
    int find_maximum_crossing_subarray(vector<int> &A, int low, int
        mid, int high) {
        // 请在这里完成你的代码
    }
    int find_maximum_subarray(vector<int> &A, int low, int high) {
        // 请在这里完成你的代码
    }
    int find_maximum_subarray(vector<int> &A) {
        return find_maximum_subarray(A, 0, A.size() - 1);
    }
};
```

测试数据范围： $N \leq 10^5$, $|A[i]| \leq 10000$ 。

2.2 循环赛日程表

设有 $n = 2^k$ 个运动员要进行羽毛球循环赛，现要设计一个满足以下要求的比赛日程表：

- 每个选手必须与其他 $n - 1$ 个选手各比赛一次
- 每个选手一天只能比赛一次
- 循环赛一共需要进行 $n - 1$ 天

一些例子：

- $n = 2$ 时的日程表如下表所示：

运动员 第一天	
1	2
2	1

- $n = 4$ 时的日程表如下表所示：

运动员	第一天	第二天	第三天
1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

- $n = 8$ 时的日程表如下表所示：

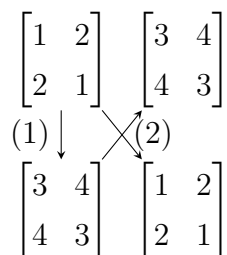
运动员	第一天	第二天	第三天	第四天	第五天	第六天	第七天
1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

你需要实现 `Solution` 类中的 `round_robin_schedule` 方法，对于传入的参数 n ，返回一个大小为 $n \times n$ 的二维数组。该二维数组满足：

- 所有元素都在 $[1, n]$ 范围内。
- 第一列从上到下依次为 $1 \sim n$ 。
- 同行或同列的元素不能相同。

```
class Solution {
public:
    vector<vector<int>> round_robin_schedule(int n) {
        // 请在这里完成你的代码
    }
};
```

提示：按照分治策略，将所有选手分成两部分。令 $m = n/2$ ，当得到了 m 名选手前 m 天的日程表后，通过平移、交换等方式分别拼凑出后 m 名选手前 m 天的日程表和 n 名选手后 m 天的日程表。下面的示意图中， $n = 4, m = 2$ 。



测试数据范围： $1 \leq k \leq 10, n = 2^k$ 。

3 实验思考

1. 对于序列 $[3, 2, -5, 3, -9, 9, -4, 6]$ ，画出求解最大子序列的递归调用树，并标出每个函数的返回值。
2. 分析两个实验的时空复杂度。
3. 如果最大子序列问题中允许选择空的子序列（和为 0），代码应该如何修改？

4. 【拓展题】完美排列（Google 面试题）

如果一个长度为 $n(1 \leq n \leq 10^3)$ 的排列 a 满足对于每对 i, j ，都不存在 k 使得 $a[k] * 2 = a[i] + a[j]$ 成立，那么该排列就被称为完美排列。给定 n ，请你求出任意一个长度为 n 的完美排列。

注意：长度为 n 的排列是指由整数 $1, 2, \dots, n$ 构成的数组。

提示：分奇偶考虑