第四天

日常膜拜

上午

一、状态压缩

Luogu P4363 一双木棋

- 小提示：状态压缩压边界线就可以
- 图解：

Luogu P5369 最大前缀和

Luogu P5492 随机算法

拓扑序计数

- 题目描述：
  > 给定一张有向无环图，求其合法拓扑序个数
  > $n \leqslant 20, m \leqslant \frac{n(n-1)}{2}$

边子集拓扑序计数

- 题目描述：
  > - 给定一张有向图（不一定是DAG）。设其边集为E
  > - 对于$T \subseteq E$定义$f(T)$为保留T时的合法拓扑序个数
  > - 求$\sum_{t \subseteq E} f(T)$
  > - $n \leqslant 20, m \leqslant n(n-1)$

Luogu P2831 愤怒的小鸟

Luogu P3959 宝藏

二、优先队列（优化DP）

经典例题：滑动窗口

- 题目描述（GitHub Copilot）：
  > 给定一个长度为n的序列，求其所有长度为k的连续子序列中的最大值
  > $n \leqslant 10^6, k \leqslant 10^5$

  > 洛谷上的题目：Luogu P1886 滑动窗口，与 GitHub Copilot 描述的题目稍有差别

- 题解（GitHub Copilot）：
  > - 用一个优先队列维护当前窗口内的最大值
  > - 每次窗口滑动时，将窗口左端点的值出队，右端点的值入队
  > - 时间复杂度$O(n \log k)$

- 实现代码（GitHub Copilot）：

```cpp
#include <iostream>
#include <queue>
using namespace std;
const int N = 1e6 + 10;
int n, k, a[N];
priority_queue<pair<int, int>> q;

int main() {
    cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> a[i];
    for (int i = 1; i <= k; i++) q.push({a[i], i});
    cout << q.top().first << ' ';
    for (int i = k + 1; i <= n; i++) {
        q.push({a[i], i});
        while (q.top().second <= i - k) q.pop();
        cout << q.top().first << ' ';
    }
    return 0;
}
```

经典例题：中位数

- 题目描述（ GitHub Copilot ）：
  > 给定一个长度为n的序列，求其所有长度为k的连续子序列中的中位数
  > $n \leqslant 10^6, k \leqslant 10^5$
- 题解（ GitHub Copilot ）：
  > - 用两个优先队列维护当前窗口内的最大值和最小值
  > - 每次窗口滑动时，将窗口左端点的值出队，右端点的值入队
  > - 时间复杂度$O(n \log k)$
- 实现代码（ GitHub Copilot ）：

```cpp
#include <iostream>
#include <queue>
using namespace std;
const int N = 1e6 + 10;
int n, k, a[N];
priority_queue<int, vector<int>, greater<int>> q1; // 小根堆
priority_queue<int> q2; // 大根堆

int main() {
    cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> a[i];
    for (int i = 1; i <= k; i++) {
        if (q2.empty() || a[i] <= q2.top()) q2.push(a[i]);
        else q1.push(a[i]);
        if (q1.size() > q2.size()) {
            q2.push(q1.top());
            q1.pop();
        }
        if (q2.size() > q1.size() + 1) {
            q1.push(q2.top());
            q2.pop();
        }
    }
    cout << (q2.size() > q1.size() ? q2.top() : (q1.top() + q2.top()) / 2) << endl;
    for (int i = k + 1; i <= n; i++) {
        if (a[i - k] <= q2.top()) q2.pop();
        else q1.pop();
        if (q2.empty() || a[i] <= q2.top()) q2.push(a[i]);
        else q1.push(a[i]);
        if (q1.size() > q2.size()) {
            q2.push(q1.top());
            q1.pop();
        }
        if (q2.size() > q1.size() + 1) {
            q1.push(q2.top());
            q2.pop();
        }
        cout << (q2.size() > q1.size() ? q2.top() : (q1.top() + q2.top()) / 2) << endl;
    }
    return 0;
}
```

简单应用：Luogu P2216 理想的正方形

Luogu P2219 修筑绿化带