

ACM Template

DUT ACM Lab

2021 年 9 月 23 日

目录

第一章 STL 使用	1
1.1 set 与 multiset	1
1.2 map 与 unordered_map	1
第二章 图论	2
2.1 二分图	2
2.1.1 二分图最大匹配	2
2.1.1.1 实现	2
2.1.1.2 性质	3
第三章 数论	4
第四章 计算几何	5
第五章 博弈理论	6
第六章 数据结构	7
第七章 典型题型	8
7.1 最长上升子序列	8
7.1.1 做法	8
第八章 杂项	9

第一章 STL 使用

1.1 set 与 multiset

1.2 map 与 unordered_map

第二章 图论

2.1 二分图

2.1.1 二分图最大匹配

2.1.1.1 实现

第一种做法

匈牙利算法，复杂度 $\Theta(nm)$

```
1  const int mx_n = 1005;
2  bool mp[mx_n][mx_n];
3  bool vis[mx_n];
4  int pre[mx_n];
5
6  bool dfs(cint loc) {
7      for(int i=n+1; i<=n+m; i++) {
8          if(mp[loc][i] && !vis[i]) {
9              vis[i] = 1;
10             if(!pre[i] || dfs(pre[i])) {
11                 pre[i] = loc;
12                 return 1;
13             }
14         }
15     }
16     return 0;
17 }
18
19 int main() {
20     int ans = 0;
21     for(int i=1; i<=n; i++) {
22         memset(vis, 0, sizeof vis);
23         ans += dfs(i);
24     }
25     cout << ans << endl;
26     return 0;
27 }
```

第二种做法

转化为网络流，复杂度依赖选择

2.1.1.2 性质

最大独立集 = n - 最大匹配

最小点覆盖 = n - 最大独立集

第三章 数论

第四章 计算几何

第五章 博弈理论

第六章 数据结构

第七章 典型题型

7.1 最长上升子序列

7.1.1 做法

第一种：

dp，复杂度 $\Theta(N^2)$ ，优点是可以记录子序列

```
1 // Nope
```

第二种：

贪心，复杂度 $\Theta(N \log N)$ ，优点是复杂度低

```
1 // Nope
```

第八章 杂项