

1 语言基础

1.1 C++ Snippets

1.1.1 define class

```
class Foo {
public:
    Foo() { this->a = 1; }
    int a;
private:
};
```

1.1.2 io

<iostream>

```
ostream &operator<<(ostream &o, Foo &f);
istream &operator>>(istream &i, Foo &f);
```

<cstdio>

```
freopen("data.in", "r", stdin);
freopen("data.out", "w", stdout);
fclose(stdin);
fclose(stdout);
```

1.1.3 random

C++ 11 <random>

```
mt19937 rng{random_device{}}();
uniform_int_distribution<int> uni(0, MAX); // [0, MAX]
int r = uni(rng);
```

C <stdlib>

```
srand(time(NULL));
int r = rand() % MAX; // [0, MAX)
```

1.1.4 other utilities

- <utility>: pair
- <bitset>

1.1.5 useful snippets

- mid of $[begin, end]$: $begin + (end - begin) / 2$
- float = float: $\text{cmath}.fabs(a - b) < 1e - 6$

2 DP

背包类. w: 背包大小, w_n: n 的重量, v_n: n 的价值,

```
// 最大价值
f(/* 0.. */ n, w) = max(
    /* 全选 */ f(n - 1, w - w_n) + v_n,
    /* 部分 */ ...,
    /* 不选 */ f(n - 1, w)
)
```

3 数学

3.1 位运算

- $n \&= n - 1$: 敲除最右边的 1 (Brian Kernighan)

3.2 数论

3.2.1 最大公约数 GCD

$gcd(a, b) \mid a > b$ (c++: <numeric>gcd)

```
func gcd(int a, b) int {
    if b == 1 {
        return a
    }
    return gcd(b, a % b)
}

// ax + by = gcd(a, b)
func exgcd(int a, b) (int x, y) {
    if b == 0 {
        x = 1; y = 0;
        return
    }
    y, x = exgcd(b, a % b)
    y -= x * (a / b)
    return
}
```

3.2.2 最小公倍数 LCM

$lcm(a, b)$ (c++: <numeric>lcm)

3.2.3 逆元

$a\%p$ 的逆元 x : $ax\%p = 1$

- $x, _ = exgcd(a, p)$
- 若 p 为素数: $x = a^{(p-2)}\%p$ (费马小定理)

应用场景

- $(n/a) \% p = nx \% p$

3.3 组合数学

3.3.1 排列 Permutation

$P(n, k) = n! / (n - k)!$

3.3.2 组合 Combination

```
C(n, k) = n! / (n - k)! / k!
        = C(n - 1, k - 1) + C(n - 1, k)
        = C(n - 1, k) * n / (n - k)
C(n, 0) = C(n, n) = 1
```

4 图论

4.1 桥

$b - f$ 是桥

```
a - b - f - g
|   |   |   |
c - d   h - i
```

4.2 欧拉图

即一笔画, 半欧拉图 (欧拉通路) 欧拉图 (欧拉回路)

4.2.1 判断

```
if (无向图) {  
    if (没有奇度节点) 欧拉图;  
    if (2个奇度节点) 半欧拉图;  
}  
if (有向图 & 强连通) {  
    if (all(入度 == 出度)) 欧拉图;  
    if (  
        only_one(abs(入度 - 出度) == 1)  
        && other(入度 == 出度)  
    ) 半欧拉图;  
}
```

4.2.2 寻找欧拉路

- Fleury: 最后走桥
- Hierholzer [[LC 332](#), [OI Wiki Euler](#)]
 1. dfs
 2. 走过后删除边
 3. 不可移动后推入栈

5 Platform

5.1 acm.ecnu.edu.cn

- `std::cin.eof()` not work?