

# Graph Encoding Challenge

EtaoinWu

# Challenge

- 将一个长度为  $\ell$  的 01 串编码为一张  $N = 100$  个点的无标号无向图，最大化  $\ell$
- 写两个程序：Alice 输入 01 串、输出无向图，Bob 输入无向图、解码出 01 串。在这二者之间，评测库会将输入的图重标号。

# Theory

- $N$  个点的无标号图有  $\approx 2^{\binom{N}{2}} / N!$  种
- 理论上限:  $\binom{N}{2} - \log N! \approx 4425$

# Basic Idea

- 01 串  $\rightarrow$  有标号无向图
- 一张有标号图中能编码  $\binom{n}{2}$  bit 的信息

# Basic Idea

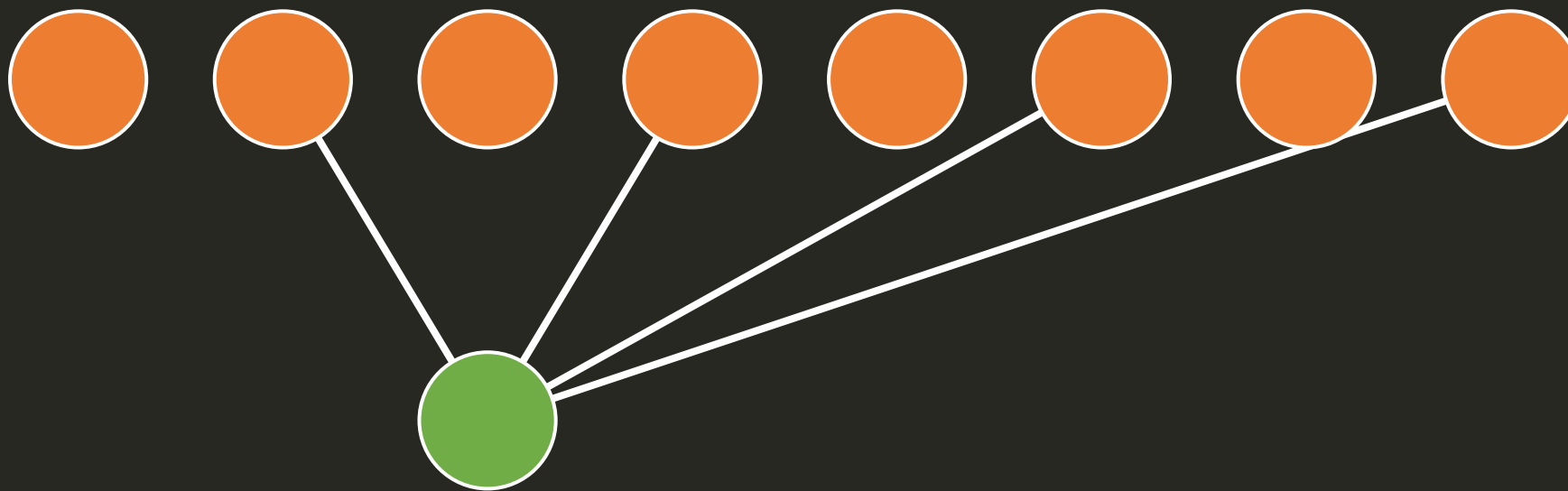
- 01 串  $\rightarrow$  有标号无向图
- 一张有标号图中能编码  $\binom{n}{2}$  bit 的信息
- 有标号无向图  $\rightarrow$  无标号无向图
- 加一些点，起到标号的作用

# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码

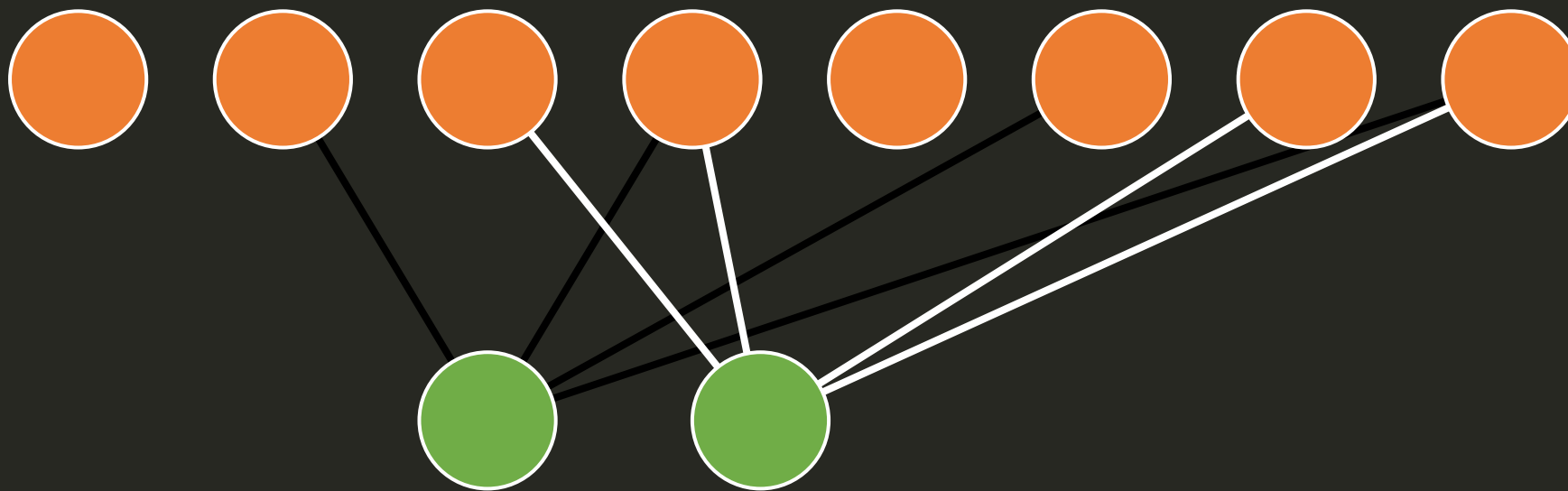
# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码



# Basic Idea

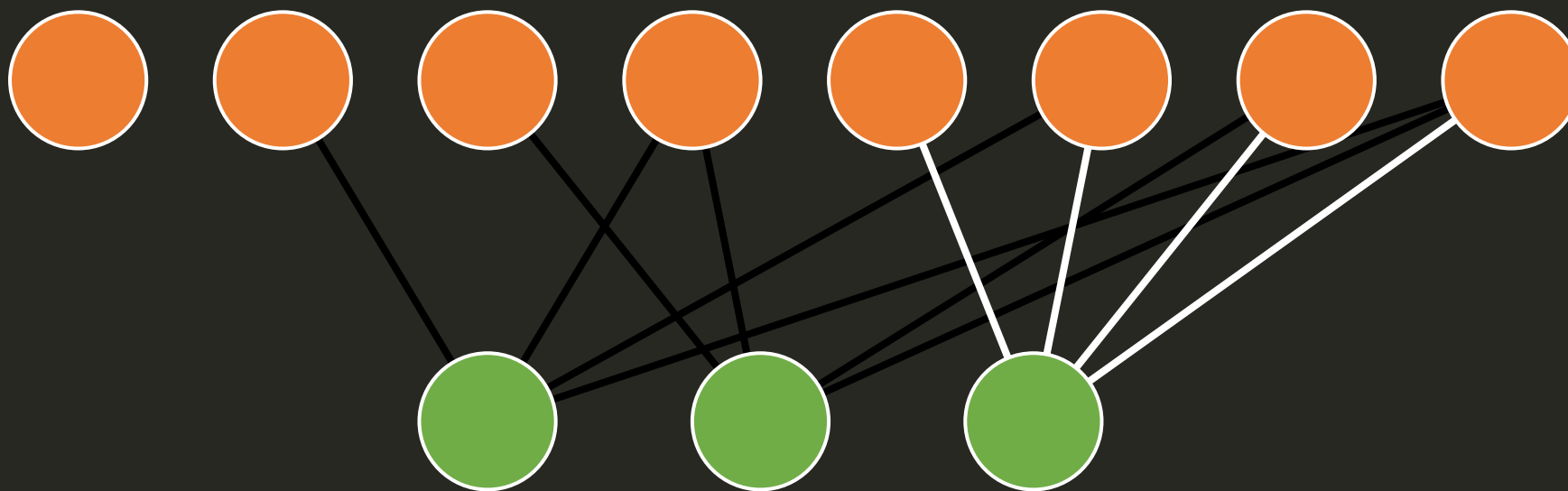
- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码





# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码



# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码



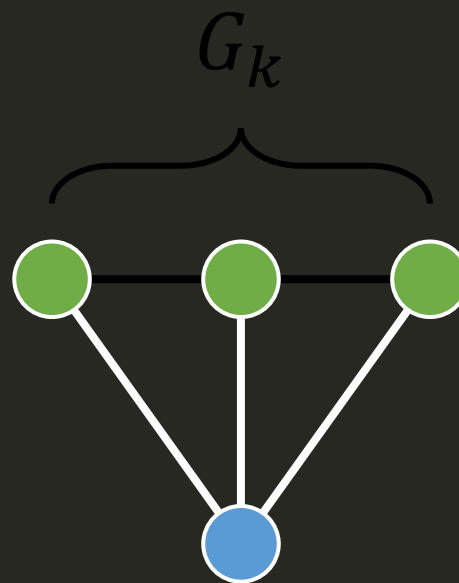
# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码



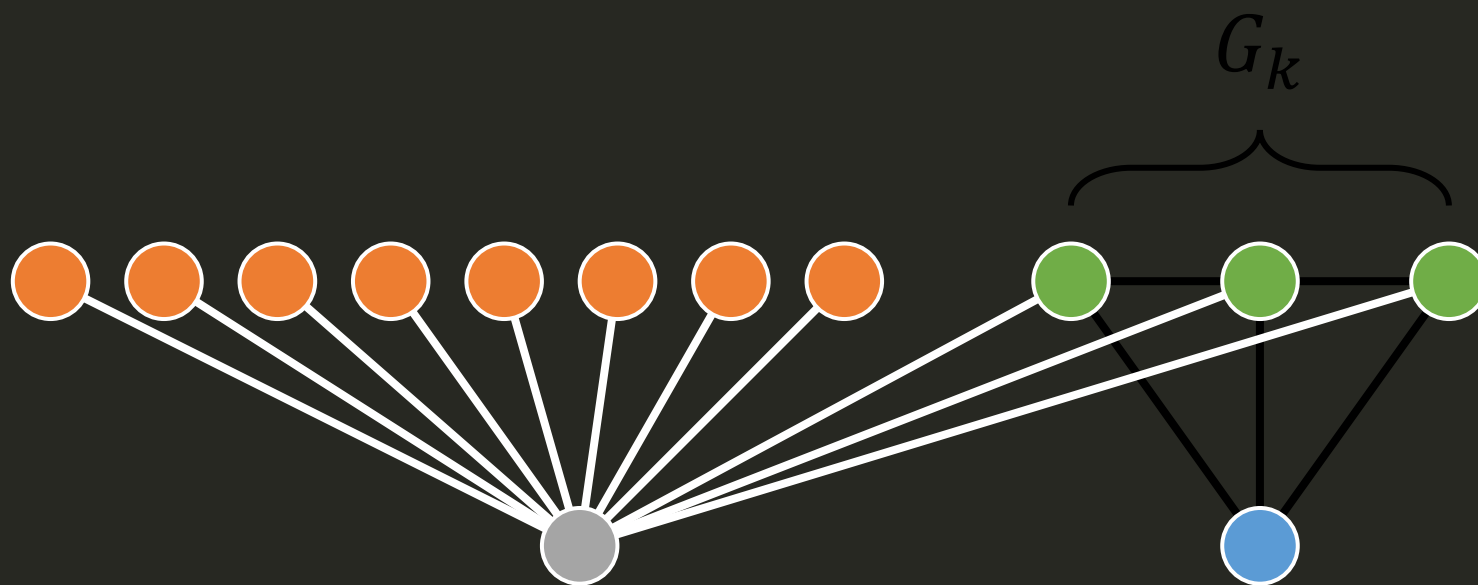
# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码



# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码

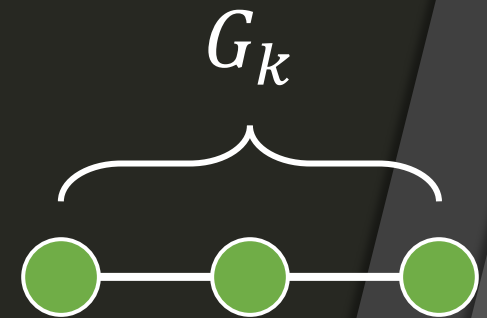


# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码
- 将  $n$  个点的有标号图编码为  $N = n + \log n + 2$  个点的无标号图

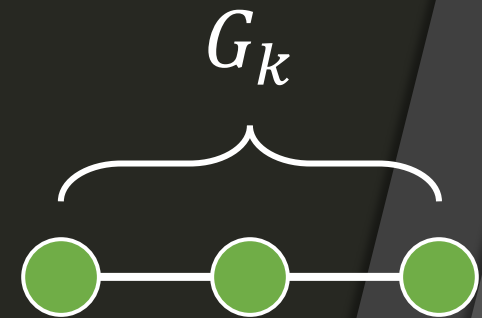
# Asymmetric Graph

- 有标号无向图  $\rightarrow$  无标号无向图
- 设计  $G_k$ ?



# Asymmetric Graph

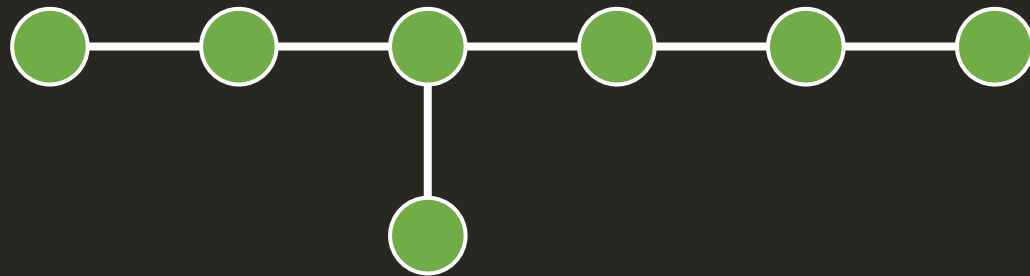
- 有标号无向图  $\rightarrow$  无标号无向图
- 设计  $G_k$ ?
- 一张没有自同构的小图
- “Asymmetric Graph”





# Asymmetric Graph

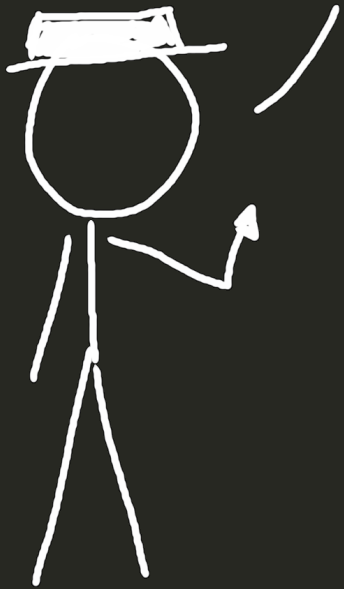
- 有标号无向图  $\rightarrow$  无标号无向图
- 设计  $G_k$ ?
- $k = 7$



# Basic Idea

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码
- 将  $n$  个点的有标号图编码为  $N = n + \log n + 2$  个点的无标号图
- $N = 100 \Rightarrow n = 91 \Rightarrow \ell = \binom{n}{2} = 4095$

Anything more exciting?



# Optimization #1

- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码
- 将  $n$  个点的有标号图编码为  $n' = n + \log n + 2$  个点的无标号图

# Optimization #1

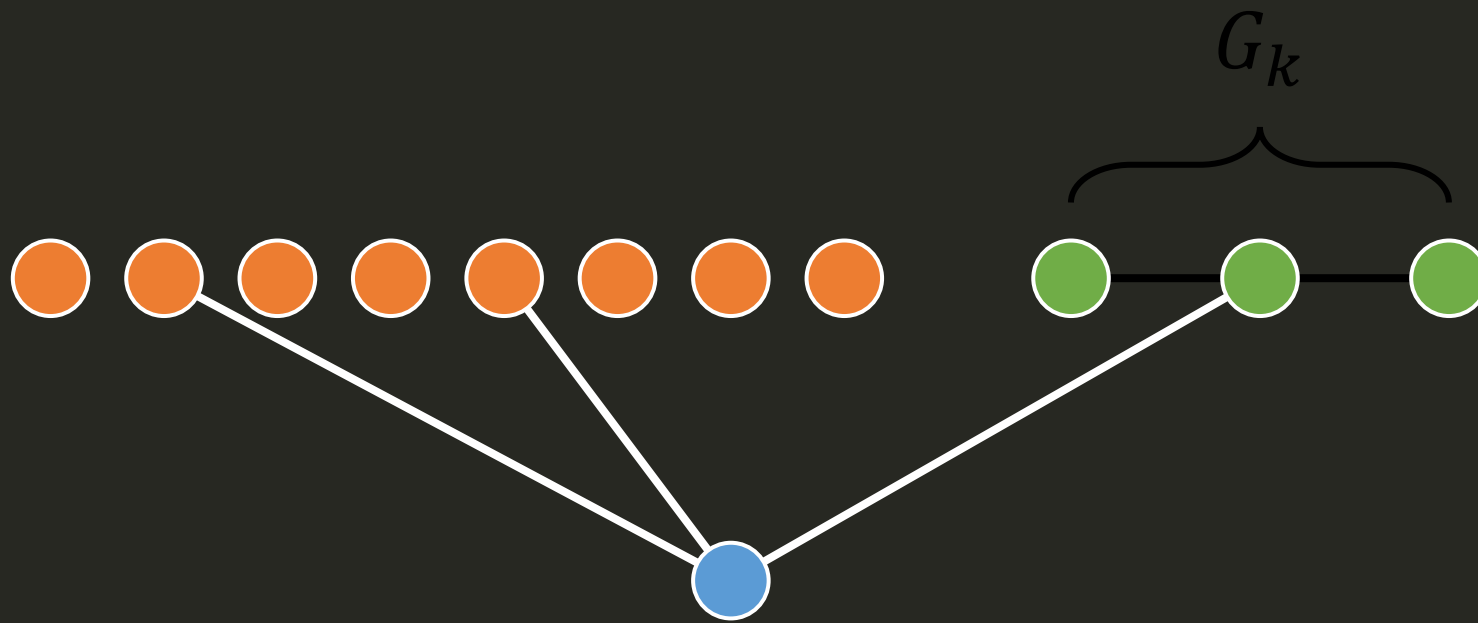
- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码
- 将  $n$  个点的有标号图编码为  $n' = n + \log n + 1$  个点的无标号图

# Optimization #1



# Optimization #1

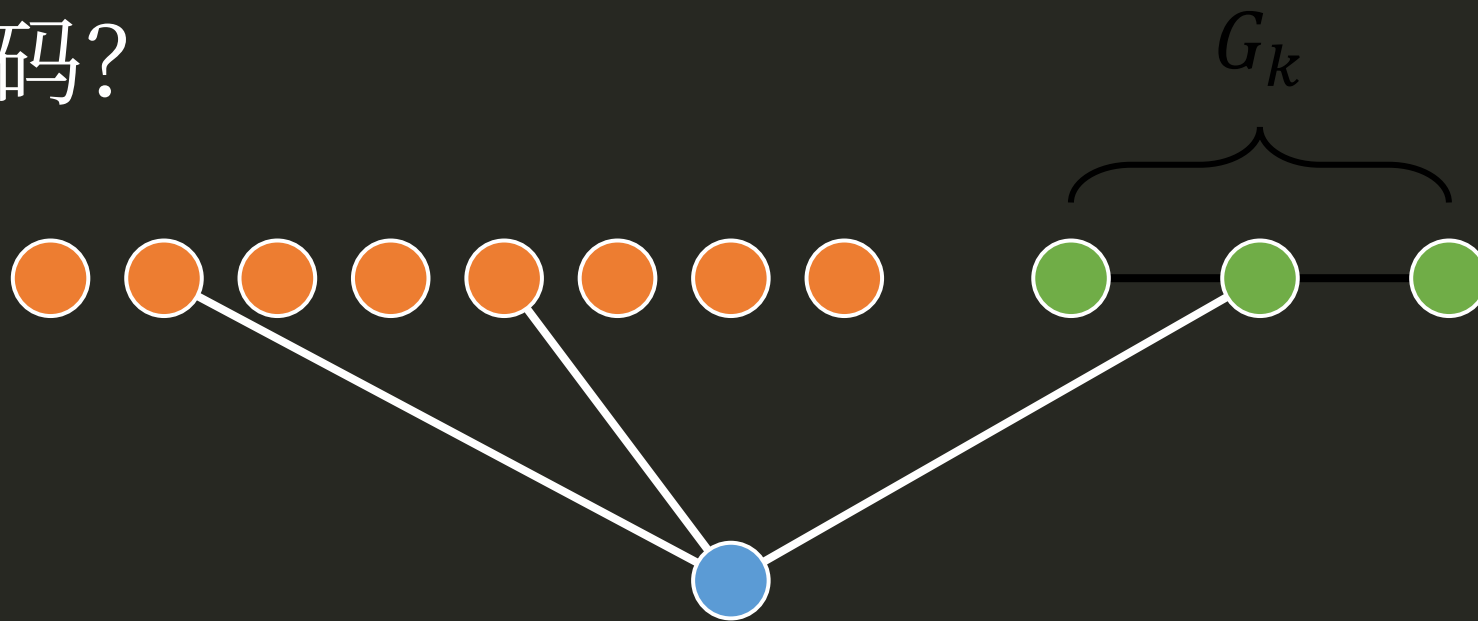
- 让所有橙色点度数为偶数、蓝绿色点度数为奇数



# Optimization #1

- 让所有橙色点度数为偶数、蓝绿色点度数为奇数

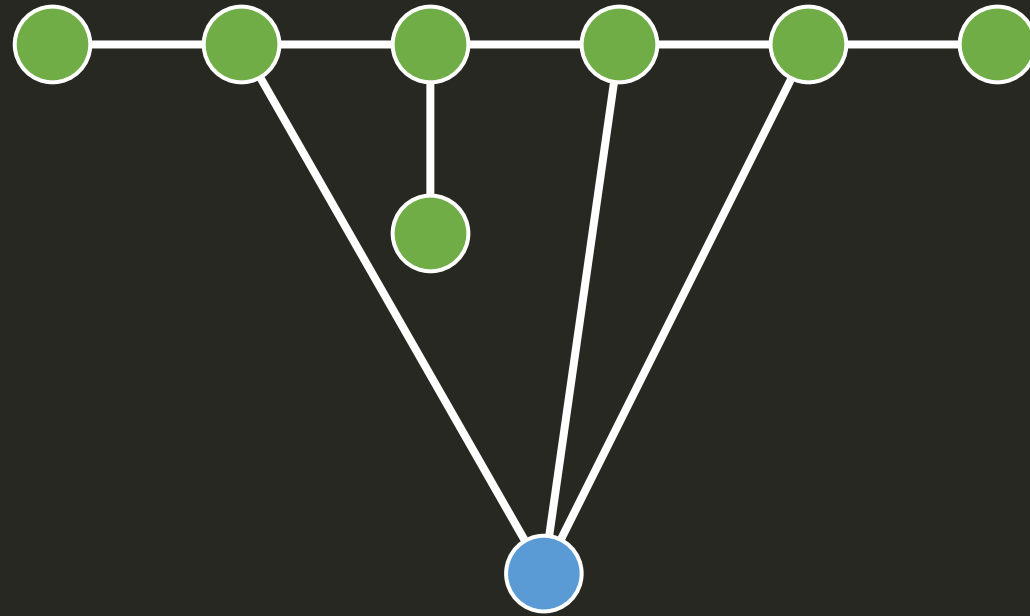
- 解码?





# Optimization #1

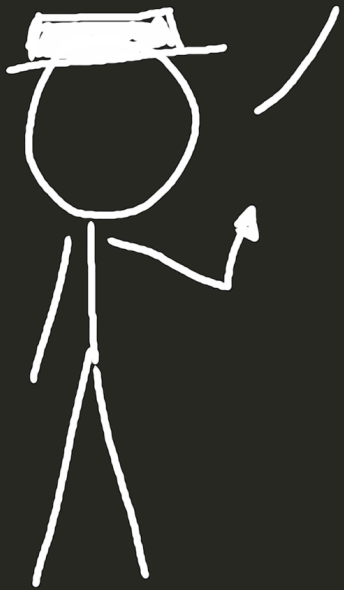
- 解码?



# Optimization #1

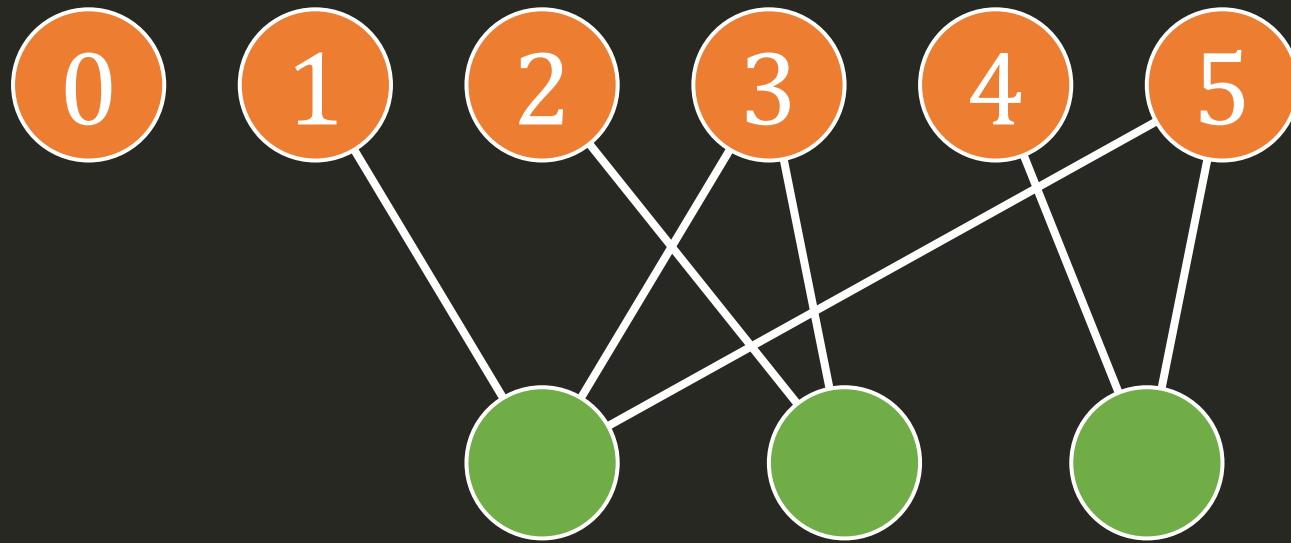
- 有标号无向图  $\rightarrow$  无标号无向图
- 二进制编码
- 将  $n$  个点的有标号图编码为  $n' = n + \log n + 1$  个点的无标号图
- $N = 100 \Rightarrow n = 92 \Rightarrow \ell = \binom{n}{2} = 4186$

Anything more exciting?



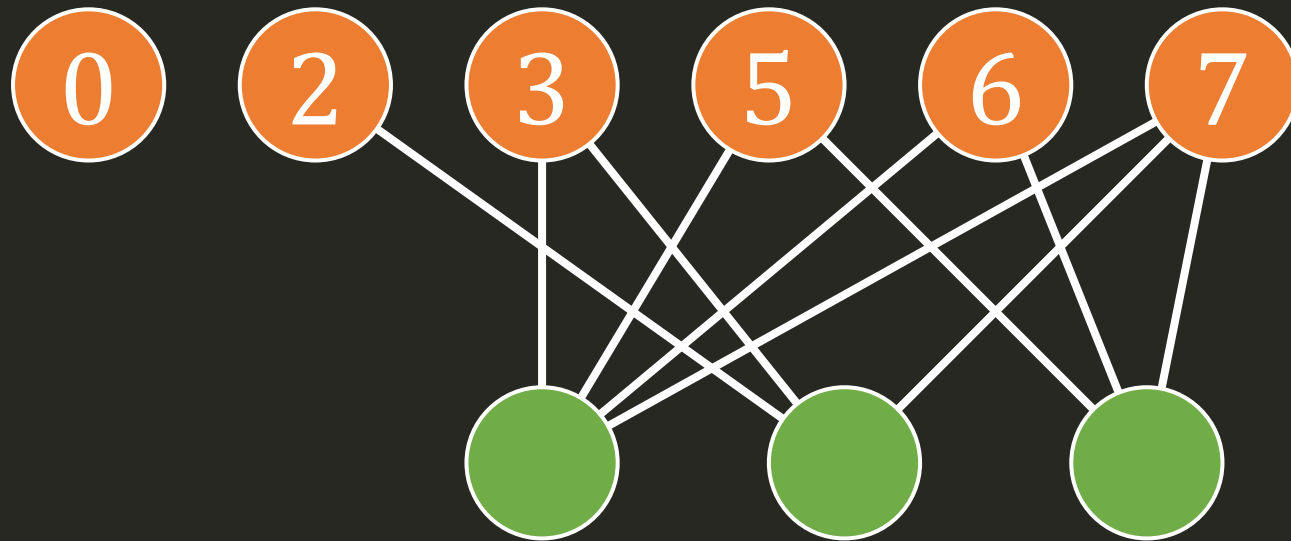
# Optimization #2

- 二进制编码



# Optimization #2

- 二进制编码

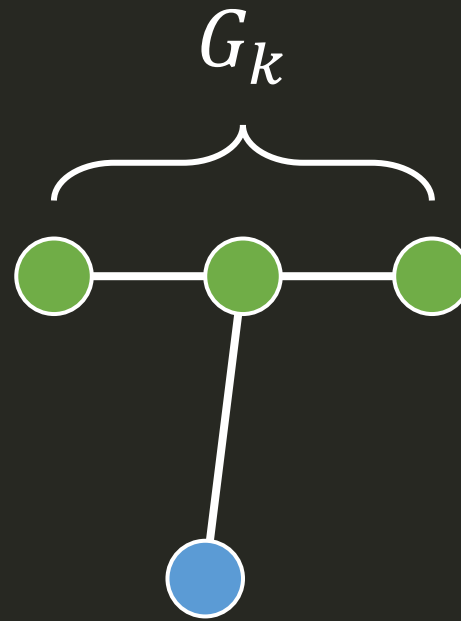


# Optimization #2

- 二进制编码
- 我们可以在  $\{0, \dots, 128\}$  中选 92 个数字分别赋给原来的点, 额外获得  $\log\binom{128}{92} \approx 106$  bit 的信息

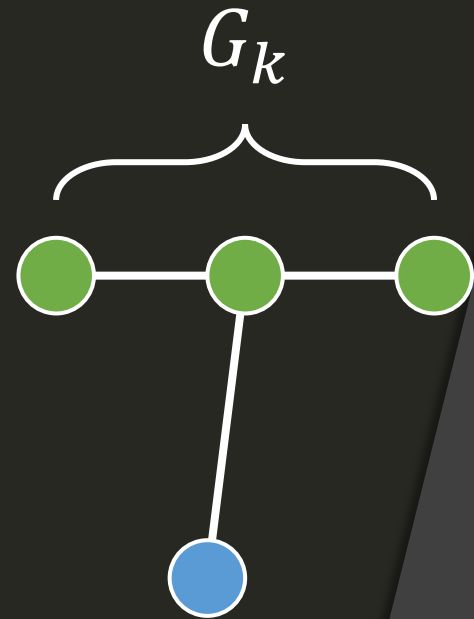
# Optimization #2

- 让所有橙色点度数为偶数、蓝绿色点度数为奇数
- 解码?



# “Error Correcting Graph”

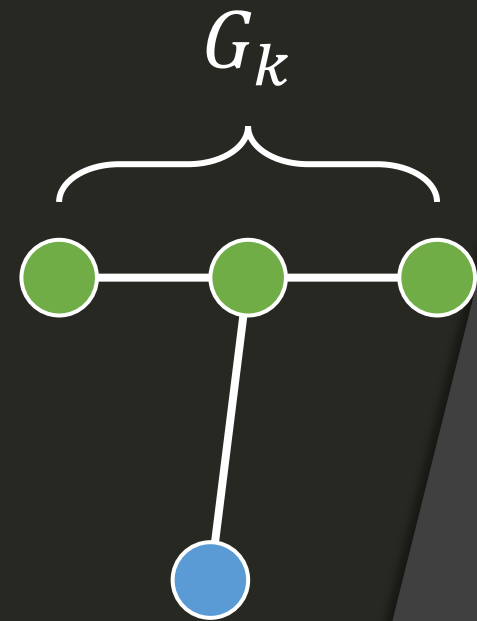
- 设计  $G_k$ ?
- 一张图，使其可以纠错一个额外的点（以及和这个点相关的一些边）





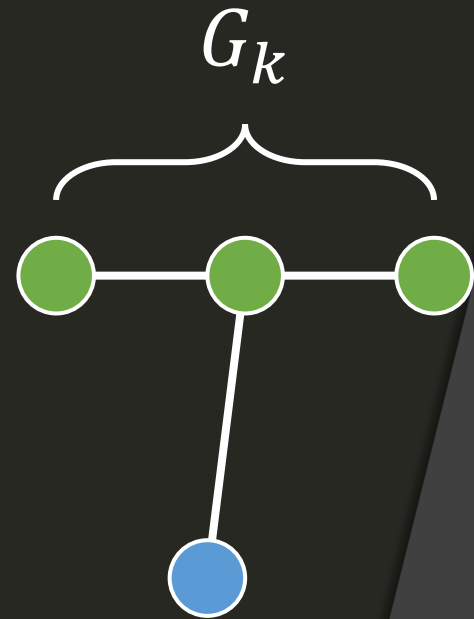
# “Error Correcting Graph”

- 设计  $G_k$ ?
- 一张图，使其可以纠错一个额外的点（以及和这个点相关的一些边）
- ... obviously does not exist.



# “Error Correcting Graph”

- 设计  $G_k$ ?
- 我们在编码的时候就知道“污染”是什么样子的。
- 可以针对“污染”设计若干张图!



# “Error Correcting Graph”

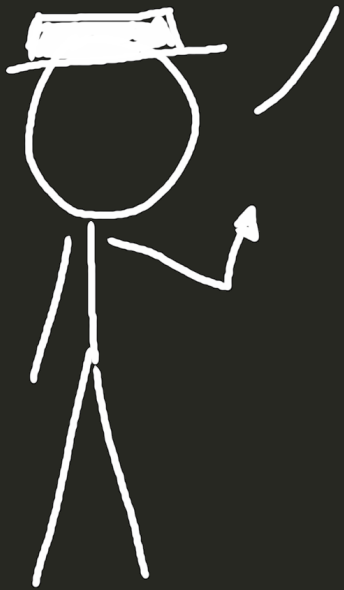
- 设计  $G_k$ ?
- 我们在编码的时候就知道“污染”是什么样子的。
- 可以针对“污染”设计若干张图!
- 在新增点度数较大的时候, 找一张稀疏图, 保证新点是度数最大的; 新增点度数较小的时候, 找一张稠密图。(略)

# Optimization #2

- 我们可以在  $\{0, \dots, 128\}$  中选 92 个数字分别赋给原来的点, 额外获得  $\log\binom{128}{92} \approx 106$  bit 的信息

- $\ell = \left\lfloor \binom{N-k-1}{2} + \log\binom{2^k}{N-k-1} \right\rfloor = 4292$

Anything more exciting?



# Optimization #3

- $\ell = \left\lceil \binom{N-k-1}{2} + \log \binom{2^k}{N-k-1} \right\rceil = 4292$

- Why is  $k = \lfloor \log n \rfloor$ ?

# Optimization #3

- $\ell = \left\lfloor \binom{N-k-1}{2} + \log \binom{2^k}{N-k-1} \right\rfloor$

- $k \in \{10, 11\}$  时  $\ell$  取最大值 4347, 我们取  $k = 11$

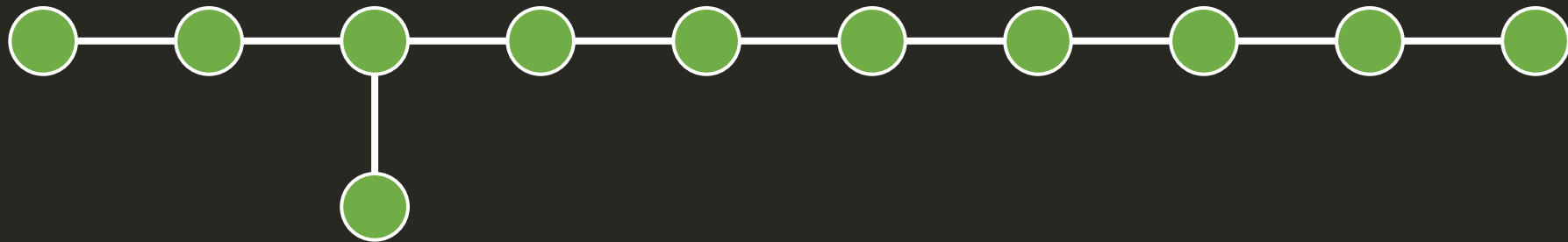
# Optimization #3

- 设计  $G_k$ ?

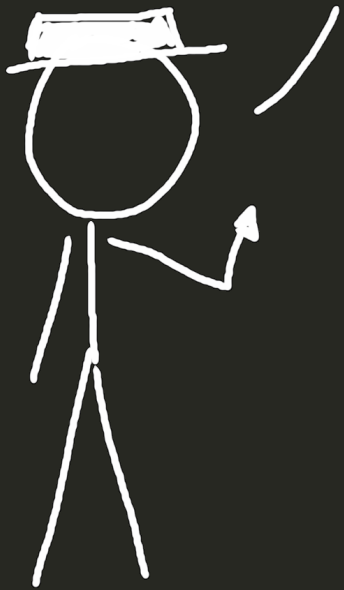


# Optimization #3

- 设计  $G_k$ ?
- 在额外点度数至少为 6 时使用下面这张图；否则使用它的补图



Anything more exciting?



# More Optimizations ...

- 整张图取补, 1 bit
- 多选几张小图, 选  $m$  张可以额外获得  $\log m$  bit  
(随手就能找到 16 张)
- 为  $k = 10$  再设计一个一样的算法, 1 bit
- ...

谢谢大家!

EtaoinWu