[07]指针、链表、杨辉三角、格雷码、GESP6 级题

XY_cpp

2025-08-10

课前须知

- 上课的时候专心听讲解，**不要跟着老师抄代码**，下课后独立完成。
- 不直接使用 AI 做题，AI 会做不等于自己会。
- 不抄袭题解（含对照题解抄一遍），抄对不等于会做。
  - 看完题解后，关闭题解独立练习。
  - 练习中途遇到问题，应当分析题目及自己的思路，而非回忆题解或再次参考题解。
- 做过的题在课后需要重新独立完成，不参考老师的课件、代码，不参考自己以前的代码。

指针

指针概念

指针也就是内存地址，指针变量是用来存放**内存地址**的变量。

为了针对不同类型的数据，指针变量也有不同的类型，比如可以有 int 类型的指针变量，其中存储的地址（即指针变量存储的数值）对应一块大小为 32 位的空间的起始地址；有 char 类型的指针变量，其中存储的地址对应一块 8 位的空间的起始地址。

指针的声明与使用

我们使用一个程序来举例：

```
01  #include <bits/stdc++.h>
02  using namespace std;
03  int main() {
04    int a = 123;
05    int* p = &a;
06    cout << p << endl;  // 0xffffc9fc903c
07    *p = 234;
08    cout << a << endl;  // 234
09
10    int b[] = {1, 2, 3};
11    p = b;
12    cout << p << " " << p[0] << endl;     // 0xffffc9fc9040 1
13    cout << ++p << " " << *p << endl;  // 0xffffc9fc9044 2
14
15    p = NULL; // 或者nullptr
16    cout << p << endl;  // 0
17    return 0;
18  }
```

new 和 delete

在 C++ 里，new 和 delete 是动态内存管理的两个关键字，用来在**堆**上申请和释放内存。

```
01  int* p = new int;          // 分配一个int
02  int* arr = new int[10];    // 分配一个int数组
03
04  delete p;        // 释放单个int
05  delete[] arr;    // 释放数组
```

- 配对使用：new 对应 delete，new[]对应 delete[]。
- 内存泄漏：如果 new 出来的指针没有 delete，**内存就不会被回收**。
- 悬空指针：delete 后，指针仍然指向那块已释放的内存，使用它会导致**未定义行为**。通常习惯写 p = nullptr; 来避免。

真题选讲

```
01  double* p_arr = new double [3];
02  p_arr[0] = 0.2;
03  p_arr[1] = 0.5;
04  p_arr[2] = 0.8;
05  p_arr += 1;
06  cout << p_arr[0] << endl;
07  p_arr -= 1;
08  delete p_arr;
```

1. 运行上面面代码，屏幕上输出是（ ）。

A. 0.2　　B. 0.5　　C. 1.2　　D. 1.5

2. (判断)最后加入语句 `cout << p_arr[1] << endl;`是未定义行为（ ）。

真题选讲

```
01  double* p_arr = new double [3];
02  p_arr[0] = 0.2;
03  p_arr[1] = 0.5;
04  p_arr[2] = 0.8;
05  p_arr += 1;
06  cout << p_arr[0] << endl;
07  p_arr -= 1;
08  delete p_arr;
```

1. 运行上面面代码，屏幕上输出是（ ）。

A. 0.2　　B. 0.5　　C. 1.2　　D. 1.5

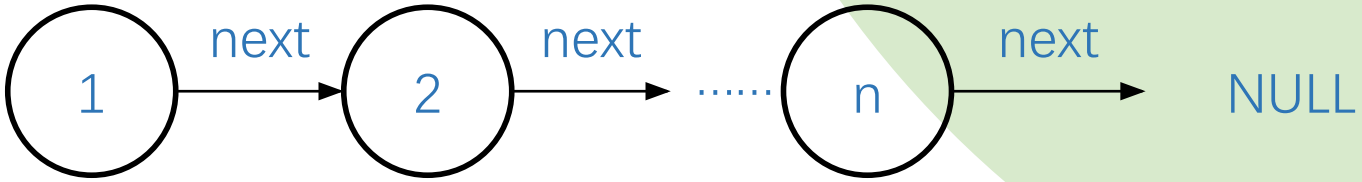
2. (判断)最后加入语句 `cout << p_arr[1] << endl;`是未定义行为（ ）。

答案：B F

链表

链表的定义

链表是一种用于存储数据的数据结构，通过如**链条一般的指针**来连接元素。



它的特点是插入与删除数据是$O(1)$的，但寻找与读取数据是$O(n)$的。

创建链表

```
01  struct Node {
02    int value;
03    Node *next;
04  };
05
06  head = new Node;
```

一般来说 head 表示链表的开头，**不存储数据**。

建议：链表的题目都用图表来辅助做题。

插入数据

1. 初始化待插入的数据 node；
2. 将 node 的 next 指针指向 p 的下一个结点；
3. 将 p 的 next 指针指向 node。

```
01  void insertNode(int i, Node *p) {
02    Node *node = new Node;
03    node->value = i;
04    node->next = p->next;
05    p->next = node;
06  }
```

删除数据

1. 将 p 下一个结点的值赋给 p，以抹掉 p->value；
2. 新建一个临时结点 t 存放 p->next 的地址；
3. 将 p 的 next 指针指向 p 的下下个结点，以抹掉 p->next；
4. 删除 t。此时虽然原结点 p 的地址还在使用，删除的是原结点 p->next 的地址，但 p 的数据被 p->next 覆盖，p 名存实亡。

```
01  void deleteNode(Node *p) {
02    p->value = p->next->value;
03    Node *t = p->next;
04    p->next = p->next->next;
05    delete t;
06  }
```

真题选讲

双向链表中每个结点有两个指针域 prev 和 next，分别指向该结点的前驱及后继结点。设 p 指向链表中的一个结点，它的前驱结点和后继结点均非空。要删除结点 p，则下述语句中错误的是（ ）。

```
01  p->next->prev = p->next;
02  p->prev->next = p->prev;
03  delete p; // A
```

```
01  p->prev->next = p->next;
02  p->next->prev = p->prev;
03  delete p; // B
```

```
01  p->next->prev = p->prev;
02  p->next->prev->next = p->next;
03  delete p; // C
```

```
01  p->prev->next = p->next;
02  p->prev->next->prev = p->prev;
03  delete p; // D
```

真题选讲

双向链表中每个结点有两个指针域 prev 和 next，分别指向该结点的前驱及后继结点。设 p 指向链表中的一个结点，它的前驱结点和后继结点均非空。要删除结点 p，则下述语句中错误的是（ ）。

```
01  p->next->prev = p->next;
02  p->prev->next = p->prev;
03  delete p; // A
```

```
01  p->prev->next = p->next;
02  p->next->prev = p->prev;
03  delete p; // B
```

```
01  p->next->prev = p->prev;
02  p->next->prev->next = p->next;
03  delete p; // C
```

```
01  p->prev->next = p->next;
02  p->prev->next->prev = p->prev;
03  delete p; // D
```

答案：A

杨辉三角

概念

如图便为杨辉三角:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

1 10 45 120 210 252 210 120 45 10 1

性质

- 除每行最左侧与最右侧的数字以外，每个数字等于它的左上方与右上方两个数字之和；
- 第 n 行的第 k 个数字为组合数$\binom{n-1}{k-1}$，即从$n-1$个物品中取出$k-1$个的方案数。

B2164

题目描述：

- 求从n个不同的元素中选择m个元素的方案数;
- 答案对10^9+7取模。

B2164

题目描述：

- 求从n个不同的元素中选择m个元素的方案数;
- 答案对10^9+7取模。

思路：

- 利用杨辉三角推导

格雷码

格雷码的定义

格雷码是一个二进制数系，其中两个**相邻数**的二进制位**只有一位不同**。举个例子，3 位二进制数的格雷码序列为：

$$000, 001, 011, 010, 110, 111, 101, 100$$

手动构造

k位的格雷码可以通过以下方法构造。我们从全 0 格雷码开始，按照下面策略：

- 翻转最低位得到下一个格雷码，（例如 $000 \to 001$）；
- 把最右边的 1 的左边的位翻转得到下一个格雷码，（例如 $001 \to 011$）；

交替按照上述策略生成 2^{k-1} 次，可得到 k 位的格雷码序列。

递归构造

构造k位格雷码的方法为：

- 递归构造$k-1$位格雷码；
- 将得到的格雷码翻转后拼接，得到：

$$G(0), G(1), ..., G\left(2^{k-1}-1\right), G\left(2^{k-1}-1\right), ..., G(1), G(0)$$

- 前一半在首位加 0，后一半在首位加 1，就得到了k位的格雷码。

真题选讲

以下代码用于生成 位格雷编码。横线上应填写（ ）。

```
01  vector<string> generateGrayCode(int n) {
02    if (n == 0) return {"0"};
03    if (n == 1) return {"0", "1"};
04    vector<string> prev = generateGrayCode(n - 1);
05    vector<string> result;
06    for (string s : prev) {
07      result.push_back("0" + s);  // 在前缀添加 0
08    }
09    for (int i = prev.size() - 1; i >= 0; i--) {
10      ____________________________  // 在此处填入代码
11    }
12    return result;
13  }
```

真题选讲

A. `result.push_back("1" + prev[i]);`

B. `result.push_back("0" + prev[i]);`

C. `result.push_back(prev[i] + "1");`

D. `result.push_back(prev[i] + "0");`

真题选讲

A. `result.push_back("1" + prev[i]);`

B. `result.push_back("0" + prev[i]);`

C. `result.push_back(prev[i] + "1");`

D. `result.push_back(prev[i] + "0");`

答案：A

GESP6 级题

洛谷

P13016

题目描述：

- 有一棵以 1 为根的树，对于编号为$k(2 \le k \le 10^9)$的结点，其父结点的编号为k的因数中除k以外最大的因数；
- q组询问，每次询问点x, y之间的距离。

P13016

题目描述：

- 有一棵以 1 为根的树，对于编号为$k(2 \le k \le 10^9)$的结点，其父结点的编号为k的因数中除k以外最大的因数；
- q组询问，每次询问点x, y之间的距离。

思路：

- 考虑求x的非自身的最大因数的复杂度为$O(\sqrt{x})$；
- 结点x的深度不会超过$\log(x)$；
- 因此求出x, y的所有祖先后暴力计算最短路径即可。

例题-P11376

题目描述：

- 题目较长，需自行阅读并理解题意
- 点击跳转

例题-P11376

思路：

- 单独考虑第i辆货车在第k个站点，那么其开销为：

$$\begin{aligned} & 2a_i p_k + 2b_i(x - p_k) \\ = & \ 2(a_i - b_i)p_k + 2b_i x \end{aligned}$$

- 贪心结论 1:
  - 若$a_i > b_i$，则p_k越小越好
  - 若$a_i < b_i$，则p_k越大越好
- 问题转换为哪辆货车**优先使用**站点

例题-P11376

- 贪心结论 2:
  - 当$a_i > b_i$时，$a_i - b_i$越大，就越应该使用p_k小的车站
  - 当$a_i < b_i$时，$a_i - b_i$越小，就越应该使用p_k大的车站
- 我们来证明$a_i > b_i$时的情况，按照这样的优先级分配时，考虑第j辆货车在第$l(k < l)$个站点，那么两者的开销为：

$$2(a_i - b_i)p_k + 2(a_j - b_j)p_l + 2(b_i + b_j)x$$

如果不按照这样的优先级分配，那么两者的开销为：

$$2(a_i - b_i)p_l + 2(a_j - b_j)p_k + 2(b_i + b_j)x$$

例题-P11376

后者减去前者，得到：

$$2\big((a_i-b_i)-(a_j-b_j)\big)(p_l-p_k)\geq 0$$

因此这样不会更优

- $a_i<b_i$时的情况同理可得

洛谷

B3873

题目描述：

- 给你N个重量为c_i，价值为l_i的物品，每件物品只能选一次，问你要获得L的价值最少需要多少重量;
- $1 \leq N \leq 500; 1 \leq L \leq 2000; 1 \leq c_i, l_i \leq 10^6$。

B3873

思路：

- 首先考虑01背包的问题：设f_j表示购买体积为j的饮料的最小花费，显然$f_j = \min(f_j, f_{j-l} + c)$；
- 问题在于j的上限是多少，如果取l的上限，显然会超时；
- 那么对于体积超过L的饮料，便可以将其体积设置为L；
- 那么就得到了j的上限为$2L$。

P10722

题目描述：

- 有一棵以 1 为根的树，结点有黑白两种颜色；
- q次操作，将以某结点为根的子树内所有节点的颜色反转；
- 最后询问每个结点的颜色；
- $n, q \leq 10^5$。

P10722

题目描述：

- 有一棵以 1 为根的树，结点有黑白两种颜色；
- q次操作，将以某结点为根的子树内所有节点的颜色反转；
- 最后询问每个结点的颜色；
- $n, q \leq 10^5$。

思路：

- 对于染色操作，先在每个结点上标记；
- 操作全部完成后 dfs 一遍二叉树，将标记向下传递；
- 传递完后当前结点的颜色即可被求出；
- 复杂度$O(n)$。