



Python 八级

2024 年 03 月

1 单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	C	D	B	A	B	C	C	B	C	D	A	A	B	D	C

第 1 题 下列代码中，用到的算法是什么算法，去掉存储的空间，算法本身用到的空间复杂度是多少（ ）

```
1 def binary_search(arr, target):
2     left = 0
3     right = len(arr) - 1
4
5     while left <= right:
6         mid = (left + right)
7
8         if arr[mid] == target:
9             return mid
10        elif arr[mid] < target:
11            left = mid + 1
12        else:
13            right = mid - 1
14
15    return -1
```

- A. 二分法， $O(\log 2N)$
- B. 二分法， $O(N)$
- C. 折半查找， $O(1)$
- D. 折半查找， $O(N \log 2N)$

第 2 题 无向图的邻接矩阵存储方法中，下列描述正确的是（ ）。

- A. 对角矩阵
- B. 稀疏矩阵
- C. 非对称矩阵
- D. 对称矩阵

第 3 题 下列代码依次输入 10,3,2 后，结果是（ ）。

```

1  def sum(n):
2      first_term = int(input())
3      common_diff = int(input())
4      last_term = first_term + (common_diff * (n - 1))
5      sum = ((first_term + last_term) / 2) * n
6      return sum
7
8  num_terms = int(input())
9  result = sum(num_terms)
10 print(result)

```

- A. 23
- B. 120
- C. 16
- D. 155

第4题 一个等边五边形，每个顶点上有一个蚂蚁，蚂蚁沿着五边形的边严格匀速行走，方向随机，请问，开始走以后，蚂蚁两两不相碰的概率是多少（ ）。

- A. 1/16
- B. 1/4
- C. 1/32
- D. 1/8

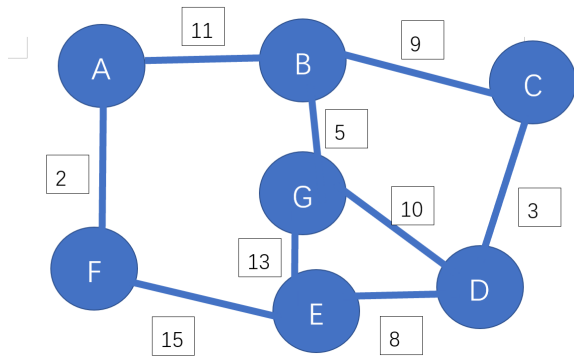
第5题 一根长度为1的小木棒，随机的折成三段，请问这三段能够组成一个三角形的概率是多少？（ ）。

- A. 1/3
- B. 1/4
- C. 1/8
- D. 1/2

第6题 有北京，雄安，天津三个城市，同样两个城市之间来回票价一样。请问火车售票部门需要准备几种车票，几种票价（ ）。

- A. 3,3
- B. 6,6
- C. 6,3
- D. 3,6

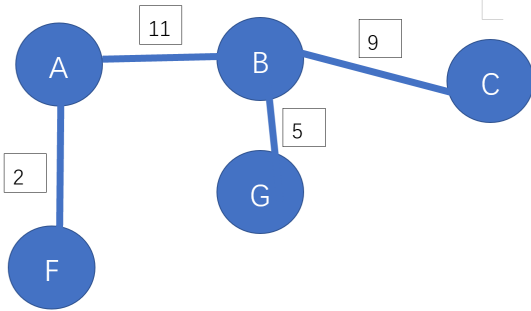
第7题 对于如下图的无向图，在用Prim算法以节点F作为起点生成最小树的过程中，哪个选项不是产生最小树的中间状态？（ ）。



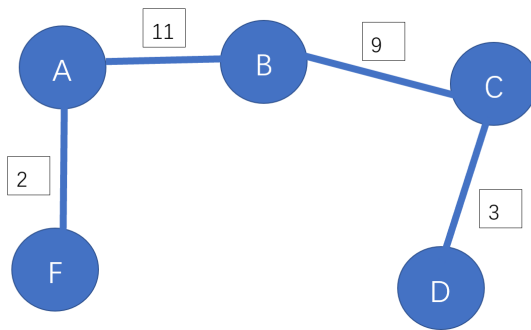
A.



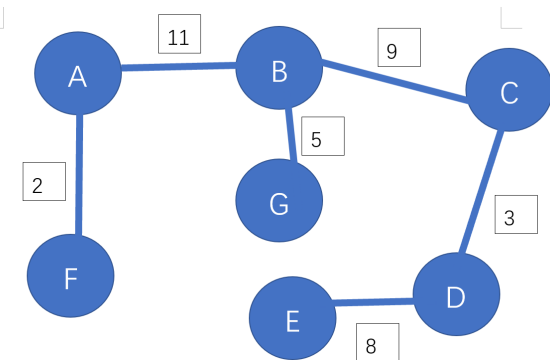
B.



C.



D.



第8题 对于一棵是完全二叉树的排序二叉树，其平均搜索的时间复杂度为（ ）。

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n^2)$

第9题 关于快速幂，下列说法错误的是（ ）。

- A. 使用了倍增思想
- B. 每一步都把指数分成两半，而相应的底数做平方运算
- C. 时间复杂度为 $O(N \log N)$
- D. 可以用快速幂方法计算斐波那契数列的第 N 项

第10题 下面实现杨辉三角形的程序中，横线处填写正确的是（ ）。

```
1 def triangles(x, y):
2     if y == 1 or y == x:
3         return 1
4     else:
5         _____
6         return z
7
```

- A. $z = \text{triangles}(x, y-1) + \text{triangles}(x, y)$
- B. $z = \text{triangles}(x-1, y+1) + \text{triangles}(x-1, y-1)$
- C. $z = \text{triangles}(x-1, y-1) + \text{triangles}(x, y)$
- D. $z = \text{triangles}(x-1, y-1) + \text{triangles}(x-1, y)$

第11题 设有编号为1, 2, 3, 4, 5的五个球和编号为1, 2, 3, 4, 5的盒子，现将这5个球投入5个盒子要求每个盒子放一个球，并且恰好有两个球的号码与盒子号码相同，问有多少种不同的方法（ ）。

- A. 20
- B. 10
- C. 12
- D. 24

第12题 1名老师和4名获奖同学排成一排照相留念，老师不站两端的排法下列所列式子正确的是（ ）。

- A. $C_3^1 A_4^4$
- B. $A_3^1 A_4^4$
- C. $C_3^1 C_4^4$
- D. $A_3^1 C_4^4$

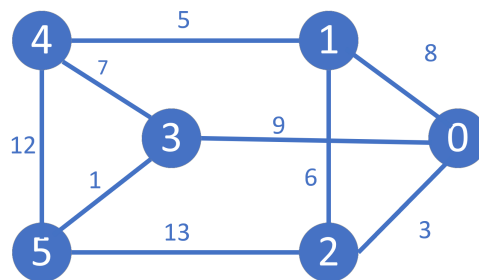
第13题 关于赋权图中，从某一个点出发，寻找最短路径的算法Dijkstra，下列说法中错误的是（ ）。

- A. 算法解决了赋权有向图或者无向图的单源最短路径问题
- B. 算法最终得到一个最短路径树
- C. 常用于路由算法或者作为其他图算法的一个子模块
- D. 算法采用的是一种贪心的策略

第 14 题 关于图的存储方法中，下列说法错误的是（ ）。

- A. 图的存储结构主要分为：邻接矩阵和邻接表
- B. 图的邻接矩阵存储方式是用两个数组来表示图：一个一维数组存储图中顶点信息，一个二维数组（邻接矩阵）存储图中的边或弧的信息。
- C. 对于边数相对顶点较少的图，邻接矩阵结构存在对存储空间的极大浪费
- D. 如果图中边的数目远远大于 n 的平方称作稀疏图，这是用邻接表表示比用邻接矩阵表示节省空间

第 15 题 Dijkstra算法中，定义S集合是已求出最短路径的节点集合，对于下图中的图，Dijkstra算法的中间形成的S集合，错误的是（ ）。



- A. $S=\{0(3)\}$
- B. $S=\{0(3),2(6)\}$
- C. $S=\{0(3),2(6),1(5)\}$
- D. $S=\{0(3),2(6),1(8)\}$

2 判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	×	√	×	×	√	√	√	×	√	√

第 1 题 线性表可以是空表，树可以是空树，图也可以是空。

第 2 题 在具有 n 个顶点、 e 条边的无向图中，无向图的全部顶点的度的和等于边数的 2 倍。

第 3 题 图的任意几个点，几个边都可以组成这个图的子图。

第 4 题 在具有 n 个顶点、 e 条边的有向图中，入度+出度的和是 $2e$ 。

第 5 题 当一棵排序二叉树退化为单支二叉树后，其平均比较次数是 $O(N)$ 。

第 6 题 不算数据的存储，插入排序算法的空间复杂度为 $O(1)$ 。

第 7 题 图的存储方式主要有两种：邻接表和邻接矩阵。

第8题 对于边数相对顶点较少的图，使用邻接矩阵来存储更好。

第9题 排列问题与顺序有关，组合问题与顺序无关。

第10题 用分治法可以优化等比数列的前 n 项求和的算法。

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：公倍数问题

3.1.1 问题描述

小 A 写了一个 $N \times M$ 的矩阵 A ，我们看不到这个矩阵，但我们可以知道，其中第 i 行第 j 列的元素 $A_{i,j}$ 是 i 和 j 的公倍数 ($i = 1, \dots, N, j = 1, \dots, M$)。现在有 K 个小朋友，其中第 k 个小朋友想知道，矩阵 A 中最多有多少个元素可以是 k ($k = 1, 2, \dots, K$)。请你帮助这些小朋友求解。

注意：每位小朋友的答案互不相关，例如，有些位置既可能是 x ，又可能是 y ，则它同时可以满足 x, y 两名小朋友的要求。

方便起见，你只需要输出 $\sum_{k=1}^K k \times \text{ans}_k$ 即可，其中 ans_k 表示第 k 名小朋友感兴趣的答案。

3.1.2 输入描述

第一行三个正整数 N, M, K 。

3.1.3 输出描述

输出一行，即 $\sum_{k=1}^K k \times \text{ans}_k$ 。

请注意，这个数可能很大，使用 C++ 语言的选手请酌情使用 `long long` 等数据类型存储答案。

3.1.4 特别提醒

在常规程序中，输入、输出时提供提示是好习惯。但在本场考试中，由于系统限定，请不要在输入、输出中附带任何提示信息。

3.1.5 样例输入 1

```
1 | 2 5 2
```

3.1.6 样例输出 1

```
1 | 9
```

3.1.7 样例解释 1

只有 $A_{1,1}$ 可以是 1，其余都不行。

$A_{1,1}, A_{1,2}, A_{2,1}, A_{2,2}$ 都可以是 2，而其余不行。

因此答案是 $1 \times 1 + 2 \times 4 = 9$ 。

3.1.8 样例输入 2

```
1 | 100 100 100
```

3.1.9 样例输出 2

```
1 | 185233
```

3.1.10 数据规模

对于 30 的测试点，保证 $N, M, K \leq 10$ ；

对于 60 的测试点，保证 $N, M, K \leq 500$ ；

对于 100 的测试点，保证 $N, M \leq 10^5$ ， $K \leq 10^6$ 。

3.1.11 参考程序

```
1 def count_divisors(limit, num):
2     s = [0] * (num + 1)
3     for i in range(1, limit + 1):
4         for j in range(i, num + 1, i):
5             s[j] += 1
6     return s
7
8 def main():
9     N, M, K = map(int, input().split())
10
11     s_N = count_divisors(N, 10 ** 6)
12     s_M = count_divisors(M, 10 ** 6)
13
14     result = 0
15     for k in range(1, K + 1):
16         result += k * s_N[k] * s_M[k]
17
18     print(result)
19
20 if __name__ == "__main__":
21     main()
```

3.2 编程题 2

- 试题名称：接竹竿

3.2.1 题面描述

小杨同学想用卡牌玩一种叫做“接竹竿”的游戏。

游戏规则是：每张牌上有一个点数 v ，将给定的牌依次放入一列牌的末端。若放入之前这列牌中已有与这张牌点数相同的牌，则小杨同学会将这张牌和点数相同的牌之间的所有牌全部取出队列（包括这两张牌本身）。

小杨同学现在有一个长度为 n 的卡牌序列 A ，其中每张牌的点数为 A_i ($1 \leq i \leq n$)。小杨同学有 q 次询问。第 i 次 ($1 \leq i \leq q$) 询问时，小杨同学会给出 l_i, r_i ，小杨同学想知道如果用下标在 $[l_i, r_i]$ 的所有卡牌按照下标顺序玩“接竹竿”的游戏，最后队列中剩余的牌数。

3.2.2 输入格式

第一行包含一个正整数 T ，表示测试数据组数。

对于每组测试数据，第一行包含一个正整数 n ，表示卡牌序列 A 的长度。

第二行包含 n 个正整数 A_1, A_2, \dots, A_n ，表示卡牌的点数 A 。

第三行包含一个正整数 q ，表示询问次数。

接下来 q 行，每行两个正整数 l_i, r_i ，表示一组询问。

3.2.3 输出格式

对于每组数据，输出 q 行。第 i 行 ($1 \leq i \leq q$) 输出一个非负整数，表示第 i 次询问的答案。

3.2.4 样例1

```
1 | 1
2 | 6
3 | 1 2 2 3 1 3
4 | 4
5 | 1 3
6 | 1 6
7 | 1 5
8 | 5 6
```

```
1 | 1
2 | 1
3 | 0
4 | 2
```

3.2.5 样例解释

对于第一次询问，小杨同学会按照 1, 2, 2 的顺序放置卡牌，在放置最后一张卡牌时，两张点数为 2 的卡牌会被收走，因此最后队列中只剩余一张点数为 1 的卡牌。

对于第二次询问，队列变化情况为：

$\{\} \rightarrow \{1\} \rightarrow \{1, 2\} \rightarrow \{1, 2, 2\} \rightarrow \{1\} \rightarrow \{1, 3\} \rightarrow \{1, 3, 1\} \rightarrow \{\} \rightarrow \{3\}$ 。因此最后队列中只剩余一张点数为 3 的卡牌。

3.2.6 数据范围

子任务编号	数据点占比	T	n	q	$\max A_i$	特殊条件
1	30	≤ 5	≤ 100	≤ 100	≤ 13	
2	30	≤ 5	$\leq 1.5 \times 10^4$	$\leq 1.5 \times 10^4$	≤ 13	所有询问的右端点等于 n
3	40	≤ 5	$\leq 1.5 \times 10^4$	$\leq 1.5 \times 10^4$	≤ 13	

对于全部数据，保证有 $1 \leq T \leq 5$ ， $1 \leq n \leq 1.5 \times 10^4$ ， $1 \leq q \leq 1.5 \times 10^4$ ， $1 \leq A_i \leq 13$ 。

3.2.7 参考程序

```
1 | import math
2 |
3 | T = int(input())
4 | for cas in range(T):
```



```

5     n = int(input())
6     m = int(math.log(n, 2)) + 1
7     A = list(map(int, input().split()))
8
9     pos = {}
10    nxt = [[n for i in range(m + 1)] for j in range(n)]
11    for i in range(n - 1, -1, -1):
12        if not (A[i] in pos):
13            nxt[i][0] = n
14        else:
15            nxt[i][0] = pos[A[i]]
16        pos[A[i]] = i
17
18    for i in range(n - 1, -1, -1):
19        for j in range(1, m + 1):
20            if nxt[i][j - 1] + 1 < n:
21                nxt[i][j] = nxt[nxt[i][j - 1] + 1][j - 1]
22
23    Q = int(input())
24    for q in range(Q):
25        l, r = map(int, input().split())
26        l, r = l - 1, r - 1
27        x, ans = l, 0
28        while x <= r:
29            while x <= r and nxt[x][0] > r:
30                x, ans = x + 1, ans + 1
31            if x > r:
32                break
33
34            for i in range(m, -1, -1):
35                if nxt[x][i] <= r:
36                    x = nxt[x][i]
37                    break
38        x += 1
39    print(ans)

```