

# CSP-S 模拟题-7

(120 分钟)

## 1. 单项选择题 (每题 2 分, 15 题, 共 30 分, 仅有一个正确选项)

1. 【NOIP2007】在 C++ 程序中, 表达式  $23 \mid 2^5$  的值是 ( )。

- A. 23      B. 1      C. 32      D. 18

2. 下列四个不同进制的数中, 与其它三项数值上不相等的是 ( )。

- A. (269)<sub>16</sub>      B. (617)<sub>10</sub>  
C. (1151)<sub>8</sub>      D. (1001101011)<sub>2</sub>

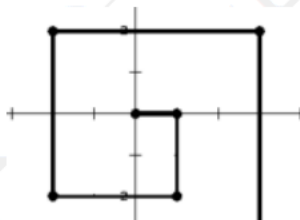
3. 有 7 个一模一样的苹果, 放到 3 个一样的盘子中, 一共有 ( ) 种放法。

- A. 7      B. 8      C. 21      D.  $3^7$

4. 现有一段文言文, 要通过二进制哈夫曼编码进行压缩。简单起见, 假设这段文言文只由 4 个汉字“之”、“乎”、“者”、“也”组成, 他们出现的次数分别为 700、600、300、200。那么, “也”字的编码长度是 ( )

- A. 1      B. 2  
C. 3      D. 4

5. 一个人站在坐标 (0, 0) 处, 面朝 x 轴正方向。第一轮, 他向前走 1 单位距离, 然后右转; 第二轮, 他向前走 2 单位距离, 然后右转; 第三轮, 他向前走 3 单位距离, 然后右...他一直这么走下去。请问第 2017 轮后, 他的坐标是: (\_\_, \_\_)。



- A. 1009, 1008      B. 1010, 1009      C. 2017, 2016      D. 1008, 1007

6. 如果开始时计算机处于小写输入状态, 现在有一只小老鼠反复按照 CapsLock、字母键 A、字母键 S、字母键 D、字母键 F 的顺序循环按键, 即 CapsLock、A、S、D、F、CapsLock、A、S、D、F、……, 屏幕上输出的第 81 个字符是字母 ( )。

- A. A      B. S      C. D      D. a

7. 根节点深度为 0, 一棵深度为 h 的满  $k$  ( $k > 1$ ) 叉树, 即除最后一层无任何子节点外, 每一层上的所有结点都有  $k$  个子结点的树, 共有 ( ) 个结点。

- A.  $(k^{h+1} - 1) / (k - 1)$   
B.  $k^{h+1}$   
C.  $k^h$   
D.  $(k^{h+1}) / (k - 1)$

8. 以下排序算法中, 不需要进行关键字比较操作的算法是 ( )。

- A. 基数排序      B. 冒泡排序      C. 堆排序      D. 直接插入排序

9. 给定一个含  $N$  个不相同数字的数组, 在最坏情况下, 找出其中最大或最小的数, 至少需要  $N - 1$  次比较操作。则最坏情况下, 在该数组中同时找最大与最小的数至少需要 ( ) 次比较操作。(  $\lceil \cdot \rceil$  表示向上取整,  $\lfloor \cdot \rfloor$  表示向下取整)

- A.  $\lceil 3N / 2 \rceil - 2$   
 B.  $\lfloor 3N / 2 \rfloor - 2$   
 C.  $2N - 2$   
 D.  $2N - 4$
10. 【NOIP2001 提高组】一棵二叉树的高度为  $h$ ，所有结点的度为 0，或为 2，则此树最少有 ( ) 个结点。  
 A.  $2^{h-1}$                       B.  $2h-1$   
 C.  $2h+1$                       D.  $h+1$
11. 由四个没有区别的点构成的简单无向连通图的个数是 ( )。  
 A. 6                      B. 7                      C. 8                      D. 9
12. 设含有 10 个元素的集合的全部子集数为  $S$ ，其中由 7 个元素组成的子集数为  $T$ ，则  $T/S$  的值为 ( )。  
 A.  $5 / 32$   
 B.  $15/128$   
 C.  $1 / 8$   
 D.  $21 / 128$
13. 10000 以内，与 10000 互质的正整数有 ( ) 个。  
 A. 2000                      B. 4000                      C. 6000                      D. 8000
14. 为了统计一个非负整数的二进制形式中 1 的个数，代码如下：
- ```
int CountBit(int x)
{
    int ret = 0;
    while(x)
    {
        ret++;
        _____;
    }
    return ret;
}
```
- 则空格内要填入的语句是 ( )。  
 A.  $x \gg= 1$   
 B.  $x \&= x - 1$   
 C.  $x |= x \gg 1$   
 D.  $x \ll= 1$
15. 一家四口人，至少两个人生日属于同一月份的概率是 ( ) (假定每个人生日属于每个月份的概率相同且不同人之间相独立)。  
 A.  $1/12$                       B.  $1/144$                       C.  $41/96$                       D.  $3/4$

## 2. 阅读程序 (程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分)

1、

```
01 #include<iostream>
02 using namespace std;
03
04 int fastpow( int a,int b,int p){
```

```

05     int ans=1;a=a%p;
06     for(int i=0;i<=31;i++){
07         if(b&(1<<i)) ans=ans*a%p;
08         a=a*a%p;
09     }
10     return ans;
11 }
12
13 int main(){
14     int a,b,p;
15     cin>>a>>b>>p;
16     cout<<fastpow(a,b,p);
17     return 0;
18 }

```

假设输入的  $a$  和  $b$  和  $p$  都是  $\text{int}$  范围内的正整数，完成下面的判断题和单选题：

•判断题：

- (1) 有必要将第 07 行和第 08 行的  $a*a$  两侧添加括号。( )
- (2) 交换第 07 行与第 08 行，答案不会改变。( )
- (3) 缩小  $b$  的范围一定不会影响代码的正确性。(指在  $a, p$  任意给出时答案仍然正确，下同，且最多缩小至 10)。( )
- (4) 缩小  $p$  的范围一定不会影响代码的正确性。( )

•选择题：

- (5) 若输入的  $a$  为 2， $b$  为 15，则输出不可能为 ( )  
 A.16068      B.16086      C.16049      D.16091
- (6) 以下说法正确的是 ( )。  
 A. 答案与  $a$  必然有相同的奇偶性。  
 B. 当输入的  $a$  变小时，输出一定变小。  
 C. 当  $a=2$  时，若  $b \leq 30$ ，则答案一定正确。  
 D. 此算法的时间复杂度为  $O(\log^2 n)$ 。

2、

```

01 #include<iostream>
02 using namespace std;
03
04 int main(){
05     int n,x;cin>>n>>x;
06     int a=0,b=0,c=0,na,nb,nc,ans=0;
07     for(int i=1;i<=n;i++){
08         int now;cin>>now;
09         na=max(a+now,0);
10         nb=max(max(a+now*x,b+now*x),0);
11         nc=max(max(c+now,b+now),0);
12         a=na,b=nb,c=nc;
13         ans=max(max(ans,a),max(b,c));
14     }
15     cout<<ans;
16 }

```

假设输入的  $n$ 、 $x$  和  $now$  都是绝对值在 1000 内的整数，完成下面的判断题和单选题：

•判断题:

- (1)一共需要输入  $n+3$  个数字。( )
- (2)这是一种朴素的未使用任何优化的动态规划算法。( )
- (3) $a$  表示前  $i$  个数的最大连续子序列和。( )
- (4)在给定的输入范围下,使用 `int` 有可能会得到错误答案。( )

•选择题:

- (5)以下说法可能是此算法对应的题目的是( )。
  - A. 给出一个长度为  $n$  的序列,可以将一段元素乘以  $x$ ,求最大连续子序列和。
  - B. 给出一个长度为  $n$  的序列,可以将不超过  $x$  个元素加上  $x$ ,求最大连续子序列和。
  - C. 给出一个长度为  $n$  的序列,可以将某些元素加上  $x$ ,求最大连续子序列和。
  - D. 给出一个长度为  $n$  的序列,求长度不超过  $x$  的最长连续子序列和。
- (6)以下输入数据得到的结果最大的是( )。
 

|                                  |                              |
|----------------------------------|------------------------------|
| A. {5, 3, 1, 2, 0, -2, 5}        | B. {3, 10, 1, -2, 1}         |
| C. {7, 1, 1, 2, -1, 1, 2, -1, 5} | D. {6, -1, 1, 2, 3, 4, 5, 6} |

3、

```
01 #include<bits/stdc++.h>
02
03 using namespace std;
04 typedef long long int_t;
05
06 int_t dis [1<<18][18];
07
08 struct E{
09     int_t to,w;
10     E(int_t to,int_t w):to(to),w(w){}
11 };
12
13 vector<E>G[20];
14
15 int_t dfs(int_t rt,int_t vided,int_t n){
16     if(rt==n-1) return 0;
17     if(dis[vided][rt]) return dis[vided][rt];
18     dis[vided][rt]=-998244353;
19     for(E e : G[rt]){
20         int_t to=e.to,w=e.w;
21         if((1<<to) & vided) continue;
22         dis[vided][rt]=max(dis[vided][rt],dfs(to,vided | (1<<to),n)+w);
23     }
24     return dis[vided][rt];
25 }
26
27 int main(){
28     int_t n,m;cin>>n>>m;
29     while(m--){
30         int_t u,v,w;cin>>u>>v>>w;
31         G[u].push_back(E(v,w));
32     }
```

```
33     cout<<dfs(0,1,n);
34 }
```

假设输入的  $n$  是不超过 18 的正整数,  $w$  不会超过 10000, 完成下面的判断题和单选题:

•判断题:

- (1) 此代码可以在 noip 标准下编译。 ( )
- (2) 此代码能处理重边与自环。 ( )
- (3) 此算法可以被 dijkstra 替代。 ( )

•选择题:

- (4) 以下说法错误的是 ( )。
  - A. 第 31 行, 为点  $u$  添加了一条边  $(v, w)$
  - B. 若去掉第 17 行, 则代码复杂度不变
  - C. 若去掉第 21 行, 则程序无法结束
  - D. 第 19 行遍历了以  $rt$  为起点的所有边

- (5) 若以下侧数据为输入数据, 则答案为 ( )。

A. 3                      B. 5                      C. 7                      D. 8

- (6) 此代码的时间复杂度为 ( )。

A.  $O(n^2+m)$                       B.  $O(2^{n+m})$                       C.  $O(2^n(n+m))$                       D.  $O(n^2 \log n+m)$

|   |   |   |
|---|---|---|
| 3 | 3 |   |
| 0 | 2 | 5 |
| 0 | 1 | 4 |
| 1 | 2 | 3 |

### 3. 完善程序题 (单选题, 每小题 3 分, 共计 30 分)

1、(贪心)你用过 QQ 吗? 在 QQ 群里, 管理员可以禁言用户。

在 Boboniu 的 QQ 群里, 小 D 每天都开 Boboniu 的玩笑。

小 D 会在群里待  $n$  天, Boboniu 的心情是  $m$ 。

在第  $i$  天, 如果小 D 没被禁言, 他会开一个严重程度为  $a_i$  的玩笑; 如果开的玩笑严重程度大于  $m$ , 他就会被 Boboniu 禁言  $d$  天, 也就是说, 在第  $i+1, i+2, \dots, \min(i+d, n)$  天, 他都会被禁言。

你可以将序列  $a$  重排, 求开的所有玩笑的严重程度之和的最大值。

【输入】

第一行是  $n, d, m$ , 之后一行  $n$  个整数  $a_i$ 。

$1 \leq d \leq n \leq 10^5, 0 \leq m \leq 10^9, 0 \leq a_i \leq 10^9$ 。

提示: 贪心。

试补全程序。

```
01 #include<bits/stdc++.h>
02 using namespace std;
03
04 const int MAXN=1e5+5;
05
06 int big[MAXN], small[MAXN], sum[MAXN];
07 int p1=1, p2=1;
08 int n, m, k, x;
09
10 int main() {
11     cin>>n>>m>>k;
12     for(int i=1; i<=n; i++) {
```

```

13         cin>>x;
14         if(x<=k){
15             ①
16         }else{
17             big[p2++]=x;
18         }
19     }
20     sort(small+1,small+1+p1,greater<int>());
21     sort(big+1,big+1+p2,greater<int>());
22     for(int i=1;i<=p1;i++){
23         ②
24     }
25     int ans=sum[p1],cur=0;
26     for(int i=1;i<=p2;i++){
27         cur+= ③;
28         int days= ④+1;
29         if(days>n){
30             break;
31         }
32         int left=min(n-days,p1);
33         ans=max(ans,⑤);
34     }
35     cout<<ans<<endl;
36     return 0;
37 }

```

(1)①处应填( )。

- A. big[++p1]=x;                      B. big[p1++]=x;  
C. small[++p1]=x;                    D. small[p1++]=x;

(2)②处应填( )。

- A. sum[i]=sum[i-1]+small[i];                      B. sum[i+1]=sum[i]+small[i+1];  
C. sum[i]=sum[i-1]+big[i];                        D. sum[i+1]=sum[i]+big[i];

(3)③处应填( )。

- A. small[i]                      B. big[i]                      C. small[i+1]                      D. big[i+1]

(4)④处应填( )。

- A. (i-1)\*m                      B. i\*(m+1)                      C. i\*m                      D. (i-1)\*(m+1)

(5)⑤处应填( )。

- A. sum[left+1]+cur                      B. sum[left]+cur  
C. sum[left-1]+cur                      D. sum[left]+cur-1

2、(动态规划)小Q在电子工艺实习课上学习焊接电路板。一块电路板由若干个元件组成，我们不妨称之为节点，并将其用数字1,2,3...进行标号。电路板的各个节点由若干不相交的导线相连接，且对于电路板的任何两个节点，都存在且仅存在一条通路（通路指连接两个元件的导线序列）。

在电路板上存在一个特殊的元件称为“激发器”。当激发器工作后，产生一个激励电流，通过导线传向每一个它所连接的节点。而中间节点接收到激励电流后，得到信息，并将该激励电流传向与它连接并且尚未接收到激励电流的节点。最终，激励电流将到达一些“终止节点”——接收激励电流之后不再转发的节点。

激励电流在导线上的传播是需要花费时间的，对于每条边e，激励电流通过它需要的时间为t，而节点接收到激励电流后



的转发可以认为是在瞬间完成的。现在这块电路板要求每一个“终止节点”同时得到激励电路——即保持时态同步。由于当前的构造并不符合时态同步的要求，故需要改变连接线的构造。目前小 Q 有一个道具，使用一次该道具，可以使得激励电流通过某条连接导线的时间增加一个单位。请问小 Q 最少使用多少次道具才可使得所有的“终止节点”时态同步？

输入第一行包含一个正整数  $N(1 \leq n \leq 5 \times 10^5)$ ，表示电路板中节点的个数。

输入第二行包含一个整数  $S$ ，为该电路板的激发器的编号。

接下来  $N-1$  行，每行三个整数  $a, b, t$ 。表示该条导线连接节点  $a$  与节点  $b$ ，且激励电流通过这条导线需要  $t(1 \leq t \leq 10^6)$  个单位时间。

提示：动态规划，代码中的 `read` 函数调用时会从输入读入一个整数，`for(X a : B)` 会枚举类型  $X$  的变量集合  $B$  中的每一个元素。

试补全程序。

```
01 #include<bits/stdc++.h>
02 using namespace std;
03
04 typedef long long int_t;
05
06 int_t read(){
07     int_t x=0,w=1;char ch=0;
08     while(!isdigit(ch)) {ch=getchar();if(ch=='-') w=-1;}
09     while(isdigit(ch)) x=①, ch=getchar();
10     return x*w;
11 }
12
13 struct E{
14     int_t to,w;
15     E(int_t to0,int_t w0) {to=to0;w=w0;}
16 };
17
18 int_t ans;
19 vector<E>G[501000];
20
21 int_t dfs(int_t rt,int_t fa){
22     int_t ret=0,cnt=0;
23     for(E e : G[rt]) if(e.to!=fa){
24         int_t res=dfs(e.to,rt)+e.w;
25         if(④) ans+=cnt*(res-ret),③,cnt++;
26         else ②,cnt++;
27     }
28     return ret;
29 }
30
31 int main(){
32     int_t n=read(),s=read();
33     for(int_t i=1;i<n;i++){
34         int_t u=read(),v=read(),w=read();
35         G[u].emplace_back(v,w);
36         G[v].emplace_back(u,w);
37     }
38     ⑤
```

```
39      cout<<ans;
40      return 0;
41 }
```

(1)①处应填 ( )。

A.  $x*10+ch-60$

B.  $x*10+ch-48$

C.  $x*10+(int)ch$

D.  $x*10+ch-65$

(2)②处应填 ( )。

A.  $ans+=(ret-res)$

B.  $ans+=(res-ret)$

C.  $ans-=(ret-res)$

D.  $ans-=(res-ret)$

(3)③处应填 ( )。

A.  $++ret$

B.  $++res$

C.  $ret+=res$

D.  $ret=res$

(4)④处应填 ( )。

A.  $res>ret$

B.  $res\geq ret$

C.  $res<ret$

D.  $res\leq ret$

(5)⑤处应填 ( )。

A.  $dfs(1, 0);$

B.  $dfs(s, 1);$

C.  $dfs(0, -1);$

D.  $dfs(s, 0);$