

CSP-S 2020 初赛试题解析

一、单项选择题

1. C

肉眼可见 C 比 A 大,而 BD 在二进制下分别是9位和10位, C 是11位.

2. B

显然.

3. B

$2048 * 1024 * 32 * 24 * 8 * 60 / 8 / 1024 / 1024 / 1024 = 90(GB)$.

4. B

$\{\} \rightarrow \{a, b\} \rightarrow \{a\} \rightarrow \{a, c, d\} \rightarrow \{a, c\}$.

5. D

A.4, 5, 1, 5.

B.4, 3, 9, 3.

C.2, 7, 10, 7.

D.1, 3, 5, 9.

6. B

B 不是(不能放得下就放), C 显然是, D 也可能是(*Dijkstra*).

A.霍夫曼编码是字符出现的次数越多编码越短的编码方法,具体编码方法自行百度.

7. A

每个点,每条边都会跑一遍,时间复杂度为 $O(n + e)$.

8. A

把所有点分成两组,同一组里不能有边,边最多的情况应为和另一组的所有点都有一条边.即求 $(24 - x)x$ 的最大值.(简单的数学题)

$(24 - x)x \leq [(24 - x) + x]^2 / 4 = 144$. (当且仅当 $x = 12$ 时取等)

9. C

显然.

10. C

CRT (中国剩余定理)解同余方程组.

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 4 \pmod{7} \end{cases}$$

当然最好是选项代入法.

11. C

求 $10k(k-1)/2 \geq 1000$ 的最小整数解.

12. D

不能不会.

13. B

$$ans = 4 * \sum_{i=1}^3 3(4-i).$$

(i 为第一个棋子所在行, 4 是列数)

14. D

每次都要找出一个到源点距离最小的点, 并用它更新剩下的点, 时间复杂度 $O(N^2)$.

15. C

香农信息论.

数学题里应该做到过香农公式: $C = W * \log_2(1 + \frac{S}{N})$.

二、阅读程序

1.

$$d_i + d_j - (d_i \& d_j) \Leftrightarrow d_i | d_j.$$

判断题

(1) ×

可以等于1000.

(2) ×

$n = 1$ 时, $ans = -1$.

(3) √

行14的 if 条件为 $d_i < d_j$, 所以 i, j 的前后顺序有影响.

若据题意更改 j 的初值, 则 if 条件应为: $d_i \neq d_j$.

(4) √

ans 的表达式中 i, j 的地位等同, 即使当前 $d_i > d_j$, 在某个循环时 i 和 j 的值会恰好调换.

单选题

(5) C

由按位或的性质可知: $a_i \leq 127$.

(6) C

输出奇数则至少有1个奇数,输出偶数则至少有2个偶数.

2.

$find(L, R, k)$ 是在 $d_{L,R}$ 中找到第 k 小的数.

还加了奇怪的快排进去.

看不出为什么的问.

判断题

(1) ×

$x \in [L, R]$.

(2) √

程序运行到行19时,根据12的 $while$ 循环条件,必有 $a = b$.

选择题

(3) 均对

单调递增,所以每层 $find()$ 中都只会在 $a = b$ 时交换一次.

交换次数与递归层数,即与随机化的 x 和 k 有关.

(4) B

在第一层递归中, d_i 会基本变成一个单调递增的序列,交换次数是 x 的期望.

而在第一层递归中, $E(x) = n/2$.

再算上递归层数带来的必然次数,平均后大约为 $O(n)$.

(5) A

对每层递归减少的区间范围求期望,大约是 $\sum_{i=1}^k n/2^i$.

平均情况下大约为 $O(n)$.

最坏情况下每层循环都只能使区间范围缩小1,时间复杂度为 $O(nk)$,即 $O(n^2)$.

(6) D

行12的 $while$ 中等号在 b 上,所以若 d_i 均相等,则必然会成为最坏情况,时间复杂度为 $O(n^2)$.

3.

笑死,完全看不懂.

先读定义的两个 $class$.

Map 其实就是 $map < string, int >$.

$Queue$ 其实就是 $queue < string >$.

行83的 for 循环条件写得很怪,但是不难看出这是一个 bfs .

$LtoR$ 函数是循环左移1位, $RtoL$ 函数是循环右移1位.

$check$ 看起来是在判断没有重复入列,并记录 bfs 的步数.

并且输出当 $s[0]$ 和 $s[1]$ 中都有 st 时的步数之和.

判断题

(1) ✓

行76, 显然.

(2) ✗

显然不行, 除非两个字符串都是对称的.

(3) ✗

因为是循环字符串, 所以**bfs**的广度只与字符串的长度有关.

*Map*在查找时, 最坏次数只取决于内在的*cnt*.

最坏情况是每次都增加2个*string*.

次数为 $2[m(m+1) + m^2 + (n-m)(n-m+1) + (n-m)^2]$.

(认为字符串比较是 $O(1)$)

用线性规划(你甚至可以肉眼观察)可以明白上述表达式最多是 $8n^2$.

所以最坏的时间复杂度是 $O(n^2)$

(或者在次数上多一些, 但绝不可能到 $O(n!)$)

选择题

(4) D

手模一下, 就会发现, 在循环移动的条件下, 互为倒序的两个字符串没有相同的时候.
所以*check*函数不会有输出.

只有在主函数的最后输出 -1 .

(5) D

$len = \{4, 6, 8\} \rightarrow ans = \{4, 14, 28\} \rightarrow delta = \{10, 14\}$.

猜想这是一个二阶等差数列.

$len = 12$ 时, $ans = 28 + 18 + 22 = 68$, 有这个选项, 很好.

别问, 问就是不会.

(6) C

神马题目, 完全不会.

由(4), 对于任意 $n \geq 3$ 的倒序字符串, 只要 $m = 0$ (偶数), 都不可能满足条件.

三、完善程序

1.

提示已经相当于让这题白给了.

从主函数开始.

行33的循环是在排序, 显然, 排序的标椎就是性价比.

故行35的*if*条件应为 $w_j/v_j < w_{j+1}/v_{j+1}$.

但是注意到D选项, 它和A在数学上等价.

而A选项事实上是整数除法, 会自动取整, 显然不符合要求.

并且即使它是实数除法, 考虑到精度问题, 我们会优先选择移项之后用乘法.

故行35为: $w[j] * v[j+1] < w[j+1] * v[j]$.

行40的 if 是在预先处理第一块蛋糕,但它显然可以和下面的循环合并.

行42的 $else$ 是蛋糕放不下的情况,所以行40的 if 是放的下情况.

故行40为: $v[1] \leq B$.

行41是把蛋糕放进去,故为: $curV = v[1]; curW = w[1]$.

行52的意义同行43,但是多了一个通分的步骤.

答案的实际值为: $curW + (B - curV) * w_i / v_i$.

结合 $print$ 函数的具体含义,故行52为: $curW * v[i] + (B - curV) * w[i], v[i]$.

行55是所有蛋糕都放进去的情况,故为: $curW, 1$.

(1) D (2) B (3) D (4) D (5) B

2.

如提示,本题是一道朴素的二进制状压DP.

从主函数开始.

行33的循环是最基本的初始化.

因为 Max 数组会存在调用未出现下标对应的值的情况,故初始化成负数.

其实可以类比成达到性背包,负数就是未到达.

行40,由提示, x 是高8位, y 是低8位.

故: $x = a >> 8$.

行41是 v 的初始化,显然, $v = 0$.

行43的循环,结合选项,易知,是在处理当前数字的高8位是 x ,上一个数字的低8位是 z 的情况.

因为高8位是确定的,所以要计算的是低8位的价值.

所以, $v = \max(v, Max[x][z] + w(y^z))$.

(to_max 函数看不懂问)

行45的循环是在处理当前数字的低8位是 y ,下一个数字的高8位是 z 的情况.

而事实上,从背包的角度来看,行42的循环是调用背包求最大值,行45的循环是更新背包维护最大值.

故, $Max[z][y] = \max(Max[z][y], v + w((x^z) << B))$.

$w(x) = x + \text{popcnt}(x)$.

w 的返回值是 s ,而 s 的初值就是 x ,所以行14的循环是在处理 x 在二进制下1的个数.

注意到 $s++$ 没有诸如 $if(x \& 1 == 1)$ 的限制条件,所以循环内每次取的都恰好是 x 的某一个1.

所以行16可以填我们比较熟悉的 $x- = x \& x - 1$,即在树状数组中每次取 x 的最低位1的操作.

而 D 和上述表达式是等价的.

(1) D (2) B (3) C (4) A (5) B