前20题有的是多选有的是单选，没有明确提示是多选还是单选，

但是可以去点击选项，可以选多个选项应该就是多选，后来知道的，

但我全选的单选。

1. 有没有问题，什么问题。

```

#include<iostream>

using namespace std;

int main(){

unique\_ptr<int> a(new int(1));

shared\_ptr<int> b(a.release());

shared\_ptr<int> c(a.get());

a.reset();

}

```

跑了下，没问题。。。

2. 不可以生成最小生成树的是 CD？

prim、Kruskal、Floyd、Dijkstra

3. 可能输出什么

```

#include<iostream>

#include<array>

using namespace std;

int main(){

std::array<int,100> a;

for(auto& i:a)

std::cout<<i<<"\n";

return 0;

}

```

a。随机输出100个不同的整数

b。100个1

c。抛出bad\_alloc异常后崩溃

d。输出100个0

下面是实际输出的：

```

0

0

0

1937330991

795698548

1919052108

796488289

1684830564

1819894831

1752391524

1684370017

1667326815

1633641832

507904

0

227520

0

48

0

16

0

0

0

37683312

1

1841378112

1

37493176

-1032945663

35618880

1

16

0

1841378144

1

37493912

1749745665

1841378160

1

37493176

622428161

35618912

1

640

0

1841378192

1

37493912

757956609

1

0

35618832

1

1841378320

1

37492160

-1788248063

0

0

-984219524

-426440337

0

0

0

0

0

0

0

0

1841378288

1

-984219524

-426440337

0

0

0

0

-984219524

-426440337

35618832

1

0

0

0

0

0

0

0

0

0

0

37683312

1

35581892

1

35618912

1

1841378400

1

37334144

```

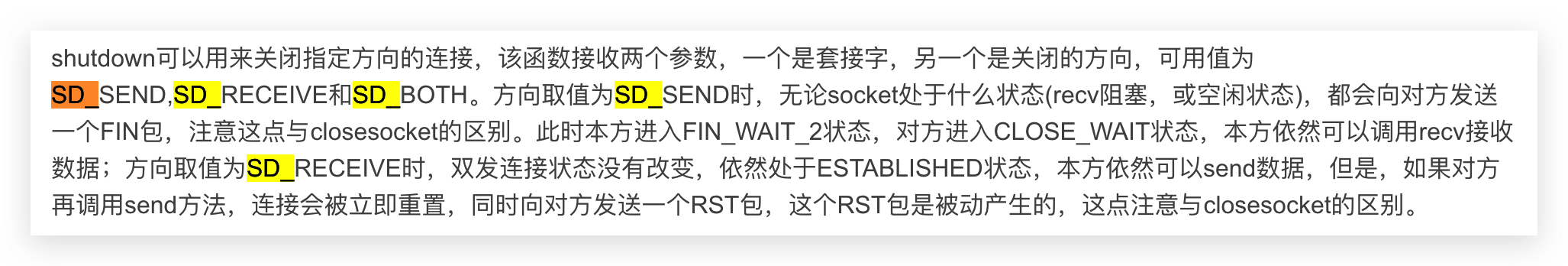
4. TCP socket 哪个套接字不会阻塞

write、read、accept会阻塞

bind、listen不会

5. TCP socket哪个阶段发FIN

send、SD\_receive、SD\_BOTH、close、

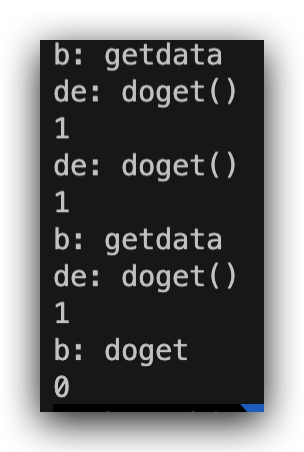


shutdown(clntSock, SD\_SEND);

6. 什么情况会用户态切换到内核态 bcd

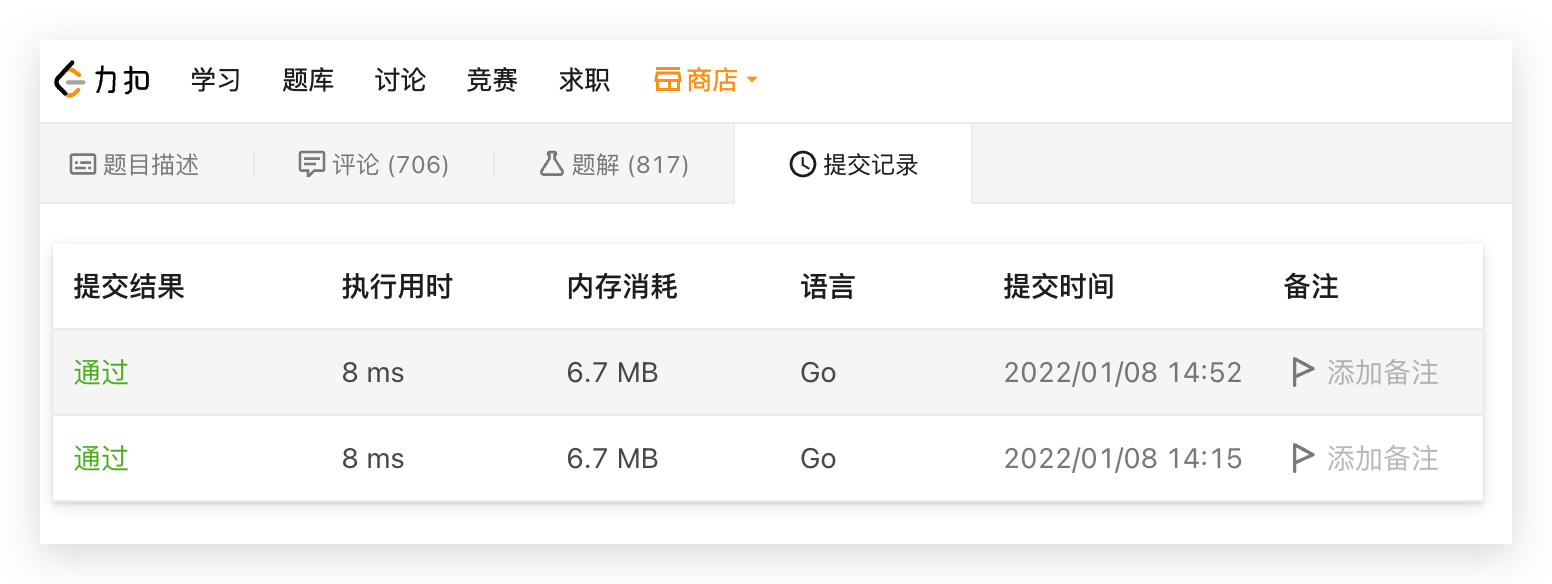
a、用户申请 b、系统调用、c、外围设备中断 d、异常

7. 输出哪4个个位数字



**第21题格雷码**

https://leetcode.cn/problems/gray-code/



```

#include<iostream>

#include<vector>

using namespace std;

// 1 0

// 11 10 00 01

// 111 110 100 101 001 000 010 011

std::vector<int> grayCode(int n){

if(n==1){

return {1,0};

}

std::vector<int> v = grayCode(n-1);

std::vector<int> ans;

for(auto i:v){

ans.push\_back((1<<(n-1))|i);

}

for(int i=v.size()-1;i>=0;i--){

ans.push\_back(v[i]);

}

return ans;

}

int main(){

int n;

cin>>n;

std::vector<int> gC = grayCode(n);

for(auto& i : gC)

cout<<i<<" ";

return 0;

}

```

不同的是力扣要求第一个数是0，这里第一个数是n位都是1的二进制数

可以做个转化在力扣上过：

```

class Solution {

public:

vector<int> f(int n){

if(n==1){

return {1,0};

}

std::vector<int> v = f(n-1);

std::vector<int> ans;

for(auto i:v){

ans.push\_back((1<<(n-1))|i);

}

for(int i=v.size()-1;i>=0;i--){

ans.push\_back(v[i]);

}

return ans;

}

std::vector<int> grayCode(int n){

vector<int> a = f(n);

vector<int> b;

int i=0;

while(a[i]!=0) i++;

int j=i;

cout<<i<<"\n";

while(j<1<<n){

b.push\_back(a[j++]);

}

j=0;

while(j<i)

b.push\_back(a[j++]);

return b;

}

};

```

\*\*第22题二维图像差值\*\*

```

#include<iostream>

using namespace std;

struct Image{

uint32\_t\* data;

uint32\_t width;

uint32\_t height;

};

// 宽/高 的比值分别是 u、v

// 两数相除只有一个比值，怎么来的两个比值

// P(result) = P(startx,starty)\*(1-u)\*(1-v) + P(startx,endy)\*(1-u)\*v + \

// P(endx,starty)\*u\*(1-v) + P(endx,endy)\*u\*v

// result 代表哪个点？

// 如果 uv是已知的常数，则 result = f(startx,starty,endx,endy)

// 说明两个点的位置确定一个终点的位置，四个点的值确定一个终点的值

// 0, 100,

// 100, 200

// 0, 50, 100,

// 50, 100, 150,

// 100, 150, 200

Image scale\_image(const Image& src, uint32\_t dst\_width, uint32\_t dst\_height){

uint32\_t sw = src.width, sh = src.height, dw = dst\_width, dh = dst\_height;

const uint32\_t\* sd = src.data;

Image dst;

dst.width = dw;

dst.height = dh;

dst.data = new uint32\_t[dw\*dh];

uint32\_t\* dd = dst.data;

int dis = (d[sw-1]-d[0])/(dw-1);

dd[0]=sd[0];

for(uint32\_t i=1;i<dw;i++)

dd[i]=dd[i-1]+dis;

int t=1;

for(uint32\_t i=dw;i<dw\*sw;i+=dw){

dd[i] = dd[t++];

}

dis=(d[sw\*sh-1]-d[sw-1])/(dh-1);

for(uint32\_t i=2\*dw-1;i<dw\*dh;i+=dw){

dd[i] = d[i-dw] + dis;

}

for(uint32\_t i = 1; i < dh; i++){

uint32\_t\* begin = dd[i\*dw];

dis = (begin[dw-1]-begin[0])/(dw-1);

for(uint32\_t j = 1; j < dw; j++)

begin[j] = begin[j-1] + dis;

}

return dst;

}

```