

知识精炼（四）



主讲人：邓哲也



Codechef FAVNUM

吉利有 N 个喜欢的数字 $a[i]$ 。

定义一个数字是吉利的，当且仅当这个数中包含至少一个吉利喜欢的数字。

比如吉利喜欢250和233，那2333就是吉利的，1250也是吉利的。

吉利想知道在 $[L, R]$ 中第 K 小的吉利数是多少。

$1 \leq L \leq R \leq 10^{18}$, $1 \leq a[i] \leq 10^{18}$, $1 \leq K \leq R-L+1$, $N \leq 62$

Codechef FAVNUM

首先做一个基本的转化。

求第 k 个满足条件的数，转化为二分答案 mid ，统计小于等于 mid 的数有多少满足条件。

联想之前做的题。

如果没有大小的限制，统计 N 位数，有多少个包含至少一个吉利的数字？

直接在 AC 自动机上 DP 即可。

这题多加了一个数字大小的限制。

那么我们可以在数位DP中加入AC自动机的状态即可。

Codechef FAVNUM

考虑建出 N 个数字串的 AC 自动机。

状态设为 $f[\text{dep}][\text{flag}][\text{status}]$

flag 表示是否有吉利的数字出现

status 表示当前走到了 AC 自动机里的那个时候节点。

枚举当前的数为 k 。

status 沿着 k 走到新的 status'

$\text{flag} \mid \text{status}'$ 的 flag 就得到了新的 flag'

也就是 $f[\text{dep}][\text{flag}][\text{status}]$ 可以从:

$$f[\text{dep}-1][\text{flag} \mid \text{trans}(\text{status}, k). \text{flag}][\text{trans}(\text{status}, k)]$$

转移来。

$\text{trans}(s, k)$ 也就是 $\text{node}[s]. \text{go}[k]$

Codechef FAVNUM

```
struct acnode{
    int go[10], fail, fl;
}s[1500];
int tot=0;
int main() {
    cin >> L >> R >> K >> N;
    memset(s, 0, sizeof(s));
    for(int i=1; i<=N; i++) {
        scanf("%s", str);
        int x=0;
        for(int j=0; str[j]; j++) {
            int c=str[j]-48;
            if(!s[x].go[c])
                s[x].go[c]=++tot;
            x=s[x].go[c];
        }
        s[x].fl=1;
    }
}
```

Codechef FAVNUM

```
int h=0, e=1;
q[0]=s[0].fail=0;
while(h<e) {
    int x=q[h++];
    s[x].f1|=s[s[x].fail].f1;
    for(int j=0; j<10; j++) {
        if(s[x].go[j]) {
            q[e++]=s[x].go[j];
            if(!x)
                s[s[x].go[j]].fail=0;
            else
                s[s[x].go[j]].fail=s[s[x].fail].go[j];
        } else s[x].go[j]=s[s[x].fail].go[j];
    }
}
```


Codechef FAVNUM

```
memset(f, -1, sizeof(f));
long long tmp=work(L-1), l=L, r=R, mid;
if(work(R)-tmp<K) printf("no such number\n");
else{
    while(l<=r){
        mid=(l+r)>>1;
        if(work(mid)-tmp>=K)    r=mid-1;
    else    l=mid+1;
    }
    cout << l << endl;
}
return 0;
}
```

Codechef FAVNUM

```
long long work(long long n) {  
    a[0]=0;  
    for(;n;n/=10)    a[++a[0]]=n%10;  
    return dp(1, a[0], 0, 0);  
}
```

Codechef FAVNUM

```
long long f[21][2][1500];

long long dp(int eq, int dep, int flag, int p) {
    if(!dep)    return flag;
    if(!eq && ~f[dep][flag][p])    return f[dep][flag][p];
    long long ret=0;
    int ed=(eq)?a[dep]:9;
    for(int i=0;i<=ed;i++)
        ret+=dp(eq&&(i==ed), dep-
1, flag||s[s[p].go[i]].f1, s[p].go[i]);
    if(!eq)    f[dep][flag][p]=ret;
    return ret;
}
```

下节课再见