

题目讨论



➤ 问题1：pencil

思路讨论：

分别讨论每种包装的笔，对于能否整除需要单独讨论一下，最后维护一个最小花费即可。

➤ 问题1：pencil

参考代码：

```
#include <bits/stdc++.h>
using namespace std;

int n, x, y, m = 10000000, s;
int main() {
    //freopen("pencil.in", "r", stdin);
    //freopen("pencil.out", "w", stdout);
    cin >> n;
    for (int i = 1; i <= 3; i++) {
        cin >> x >> y;
        if (n % x == 0) s = n / x * y;
        else s = (n / x + 1) * y;
        m = min(m, s);
    }
    cout << m << endl;
    return 0;
}
```

➤ 问题2：date

思路讨论：

枚举日期的后半部分，然后颠倒(reverse)一下，然后跟给定日期的年份进行判断一下即可计数。

对于闰年的情况，其实涉及到是闰年又是回文的只有92200229这种情况，单独特殊判断一下即可。

➤ 问题2：date

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
int n, m, a, p;
int yue[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
int main() {
    freopen("date.in", "r", stdin);
    freopen("date.out", "w", stdout);
    scanf("%d%d", &n, &m);
    for(int i = 1; i <= 12; i++) {
        for(int j = 1; j <= yue[i - 1]; j++) {
            a = (j % 10) * 1000 + (j / 10) * 100 + (i % 10) * 10 + i / 10;
            if(a < (n / 10000) || a > (m / 10000)) continue;
            p++;
        }
    }
    if(n <= 92200229 && m >= 92200229) p++;
    printf("%d\n", p);
    return 0;
}
```

➤ 问题3：port

思路讨论：

40p做法：

直接暴力统计信息

时间复杂度 $O(n^2)$

100p做法：

维护一个长度为24h的时间窗口，每次有新的船只来的时候，更新信息和删除已经失效的信息。具体实现看代码。

时间复杂度 $O(n)$

➤ 问题3：port

参考代码：

```
#include <bits/stdc++.h>
using namespace std;

int n, l, r, ans, tot, f;
int s[100005], t[100005], p[100005], q[300005];
//s数组为统计每种国籍的数量
//t数组为每艘船到达的时间点
//p数组为每艘船上的人数
//q数组为每个人的国籍
int main() {
    scanf("%d", &n);
    for(int i = 1; i <= n; i++) {
        r++;
        scanf("%d%d", &t[i], &p[i]);
        for (int j = 1; j <= p[i]; j++) {
            tot++;scanf("%d", &q[tot]);s[q[tot]]++;
            if (s[q[tot]] == 1) ans++;
        }
        while(t[r] - t[l] >= 86400) {
            for(int j = 1; j <= p[l]; j++) {
                f++;s[q[f]]--;
                if(s[q[f]] == 0) ans--;
            }
            l++;
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

➤ 问题4：magic

思路讨论：

直接按题意四层循环枚举，期望得分40分。

注意 n 这个变量我们在暴力中并没有用上，所以这个题还需要进一步分析。

➤ 问题4：magic

思路讨论：

仔细分析我们要满足的条件：

$X_a < X_b < X_c < X_d$ ， $X_b - X_a = 2(X_d - X_c)$ ，且 $X_b - X_a < (X_c - X_b)/3$ ，

我们把数量关系用图表示出来：



➤ 问题4：magic

思路讨论：

于是我们可以枚举cd的距离，有了cd的距离，我们再枚举d的魔法值。

d的方案数 = (前面所有a的方案)*(前面所有b的方案)*(当前c的方案数)

c的方案数 = (前面所有a的方案)*(前面所有b的方案)*(当前d的方案数)

同理我们可以枚举a的位置，也可以得到a和b的方案数。

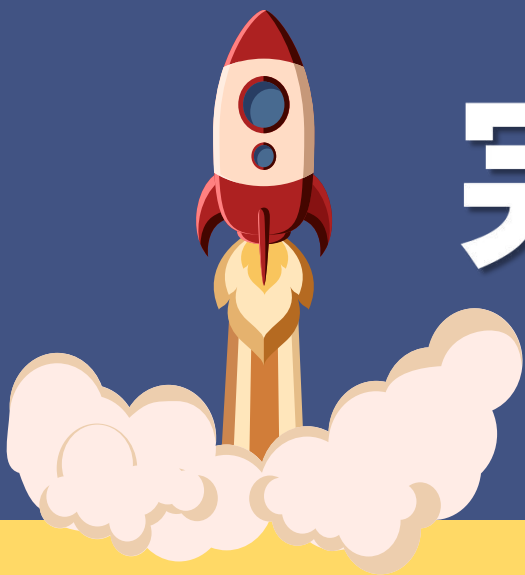
➤ 问题4：magic

参考代码：

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 15000 + 5;
const int maxm = 40000 + 5;

int n, m;
int A[maxn], B[maxn], C[maxn], D[maxn], f[maxn], cnt[maxn];
int main() {
    scanf("%d%d", &m, &n);
    for(int i = 1; i <= n; ++i) {
        scanf("%d", &f[i]);
        ++cnt[f[i]];
    }
    for(int t = 1; 9 * t <= m; ++t) {
        int sum = 0;
        for(int d = t * 9 + 2; d <= m; ++d) {
            int c = d - t, b = d - 7 * t - 1, a = d - 9 * t - 1;
            sum += cnt[a] * cnt[b];
            C[c] += cnt[d] * sum;
            D[d] += cnt[c] * sum;
        }
        sum = 0;
        for(int a = m - t * 9 - 1; a >= 1; --a) {
            int b = a + 2 * t, c = a + 8 * t + 1, d = a + 9 * t + 1;
            sum += cnt[c] * cnt[d];
            A[a] += cnt[b] * sum;
            B[b] += cnt[a] * sum;
        }
    }
    for(int i = 1; i <= n; ++i) printf("%d %d %d %d\n", A[f[i]], B[f[i]], C[f[i]], D[f[i]]);
    return 0;
}
```



完！

以梦为码 心之所往

