

福建师范大学24届低年级程序设计大赛题解

A: 小胖吹气球

签到题，答案是 $n * x + n * x / k - (n * x \% k == 0)$

```
1 #include <stdio.h>
2 #include <assert.h>
3
4 int main() {
5     int T;
6     scanf("%d", &T);
7     while (T--) {
8         int n, x, k;
9         scanf("%d %d %d", &n, &x, &k);
10        printf("%d\n", n * x + (n * x / k) - (n * x % k == 0));
11    }
12    return 0;
13 }
```

B: 小胖寻路

签到题，可以发现节点 u 的父节点为 $\lfloor \frac{u}{2} \rfloor$ ，所以只要计算 u 除几次 2 能到根节点 1 即可。或者 $\log_2(n)$ 向下取整。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int T, n;
5 int main() {
6     scanf("%d", &T);
7     while(T--) {
8         scanf("%d", &n);
9         int ans = 0;
10        while(n != 1) {
11            ans++;
12            n /= 2;
13        }
14        printf("%d\n", ans);
15    }
16    return 0;
17 }
```

C: 小胖合并石子

显然，把所有石子放到所有石子位置的中位数是最优解。因为 a_i 比较小，所以也可以预处理前缀和，枚举每个位置取 \min 。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 typedef long long ll;
5 const int maxn = 1e6 + 100;
6
7 int T;
8 ll n, a[maxn];
9 int main() {
10    scanf("%d", &T);
11    while(T--) {
12        scanf("%lld", &n);
13        for(int i = 1; i <= n; i++) {
14            scanf("%lld", &a[i]);
15        }
16        sort(a+1, a+1+n);
17        ll x = a[(n+1)/2], ans = 0;
18        for(int i = 1; i <= n; i++) {
```

```

19     ans += abs(a[i] - x);
20     }
21     printf("%lld\n", ans);
22     }
23     return 0;
24 }

```

D: 小胖斗地主

这题就是考验代码的模拟能力。这题的难度和做法有关。这里我们推荐一种简单的模拟方式，就是讲 A 和 2 转化成 14 和 15 进行模拟即可，注意特判炸弹的情况即可。

代码1:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  const int N = 100 + 5, M = 15;
5
6  int n, op, k, fg, ph[N], xp[N], buc[N];
7
8  int BOOM() {
9      for (int i = M; i >= 3; i--) if (buc[i] == 4) {
10         return i;
11     }
12     return 0;
13 }
14
15 void solve1() { //炸弹
16     int mx = BOOM();
17     if (mx > ph[1]) {
18         fg = 1;
19     }
20 }
21
22 void solve2() { //单牌
23     for (int i = ph[1] + 1; i <= M; i++) if (buc[i]) {
24         fg = 1;
25     }
26     if (BOOM()) {
27         fg = 1;
28     }
29 }
30
31 void solve3() { //对子
32     for (int i = ph[1] + 1; i <= M; i++) if (buc[i] >= 2) {
33         fg = 1;
34     }
35     if (BOOM()) {
36         fg = 1;
37     }
38 }
39
40 void solve4() { //三带一
41     int s = 0;
42     if (ph[1] == ph[3]) s = ph[1];
43     else if (ph[2] == ph[4]) s = ph[4];
44     assert(s != 0);
45     for (int i = s + 1; i <= M; i++) if (buc[i] == 3 && n >= 4) {
46         fg = 1;
47     }
48     if (BOOM()) {
49         fg = 1;
50     }
51 }
52
53 void solve5() { //三带二
54     int s = 0;
55     if (ph[1] == ph[3]) s = ph[1];
56     else if (ph[3] == ph[5]) s = ph[5];
57     assert(s != 0);
58     for (int i = s + 1; i <= M; i++) if (buc[i] == 3) {
59         for (int j = 3; j <= M; j++) {
60             if (j == i) continue;
61             if (buc[j] >= 2) {
62                 fg = 1;
63             }
64         }
65     }

```

```

66     if (BOOM()) {
67         fg = 1;
68     }
69 }
70
71 void solve6() { // 顺子
72     for (int i = ph[1] + 1; i <= M; i++) {
73         int res = 0;
74         for (int j = i; j <= min(M - 1, i + k - 1); j++) {
75             res += (buc[j] != 0);
76         }
77         if (res == k) {
78             fg = 1;
79             break;
80         }
81     }
82     if (BOOM()) fg = 1;
83 }
84
85 int main() {
86     int T;
87     cin >> T;
88     while (T--) {
89         cin >> op >> k;
90         for (int i = 1; i <= k; i++) {
91             cin >> ph[i];
92             if (ph[i] == 1) ph[i] = 14;
93             if (ph[i] == 2) ph[i] = 15;
94         }
95         memset(buc, 0, sizeof buc);
96         cin >> n;
97         for (int i = 1; i <= n; i++) {
98             cin >> xp[i];
99             if (xp[i] == 1) xp[i] = 14;
100            if (xp[i] == 2) xp[i] = 15;
101            buc[xp[i]]++;
102        }
103        sort(ph + 1, ph + 1 + k);
104        sort(xp + 1, xp + 1 + n);
105        fg = 0;
106        switch(op) {
107            case 1: solve1(); break;
108            case 2: solve2(); break;
109            case 3: solve3(); break;
110            case 4: solve4(); break;
111            case 5: solve5(); break;
112            case 6: solve6(); break;
113        }
114        if (fg) {
115            cout << "yes" << '\n';
116        } else {
117            cout << "no" << '\n';
118        }
119    }
120    return 0;
121 }

```

代碼2:

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int T;
6  int n, m, typ, a[20];
7  int check() {
8      for(int i = 15; i >= 1; i--) {
9          if(a[i] >= 4) return 16 + i;
10     }
11     if(typ == 2) {
12         for(int i = 15; i >= 1; i--) {
13             if(a[i]) return i;
14         }
15     }
16     else if(typ == 3) {
17         for(int i = 15; i >= 1; i--) {
18             if(a[i] >= 2) return i;
19         }
20     }
21     else if(typ == 4) {
22         for(int i = 15; i >= 1; i--) {
23             if(a[i] >= 3) {
24                 for(int j = 1; j <= 15; j++) {

```

```

25         if(i == j) continue;
26         if(a[j]) return i;
27     }
28     }
29 }
30 }
31 else if(typ <= 5) {
32     for(int i = 15; i >= 1; i--) {
33         if(a[i] >= 3) {
34             for(int j = 1; j <= 15; j++) {
35                 if(i == j) continue;
36                 if(a[j] >= 2) return i;
37             }
38         }
39     }
40 }
41 else {
42     int pre = 14;
43     for(int i = 14; i >= 1; i--) {
44         if(!a[i]) pre = i-1;
45         if(a[i] && pre - i + 1 >= n) return pre;
46     }
47 }
48 return 0;
49 }
50
51 int main() {
52     scanf("%d", &T);
53     while(T--) {
54         scanf("%d%d", &typ, &n);
55         for(int i = 1; i <= 16; i++) a[i] = 0;
56         for(int i = 1, x; i <= n; i++) {
57             scanf("%d", &x);
58             if(x <= 2) x += 13;
59             a[x]++;
60         }
61         int ans1 = check();
62         for(int i = 1; i <= 16; i++) a[i] = 0;
63         scanf("%d", &m);
64         for(int i = 1, x; i <= m; i++) {
65             scanf("%d", &x);
66             if(x <= 2) x += 13;
67             a[x]++;
68         }
69         int ans2 = check();
70         puts(ans1 < ans2 ? "yes":"no");
71     }
72     return 0;
73 }

```

E: 小胖玩游戏

贪心，保证每一天得到的经验值最大即为最优解。我们可以考虑优先队列，把这一天可以得到的经验用一个优先队列维护，每天取出最大的经验即可。

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int N = 2e5 + 5;
5
6 int n, m, now;
7
8 struct node {
9     int p, a;
10     bool operator < (const node &o) const {
11         return p < o.p;
12     }
13 } w[N];
14
15 int main() {
16     int T;
17     scanf("%d", &T);
18     while (T--) {
19         scanf("%d %d", &n, &m);
20         for (int i = 1; i <= m; i++) {
21             scanf("%d %d", &w[i].p, &w[i].a);
22         }
23         sort(w + 1, w + 1 + m);
24         now = 1;

```

```

25     long long ans = 0;
26     priority_queue<int> q;
27     for (int i = 1; i <= n; i++) {
28         while (now <= m && w[now].p <= i) q.push(w[now].a), ++now;
29         if (!q.empty()) {
30             ans += q.top();
31             q.pop();
32         }
33     }
34     printf("%lld\n", ans);
35 }
36 return 0;
37 }

```

F: 小胖打怪

因为要最小化值，考虑二分。对于每个二分出的答案，我们可以优先使用冲撞，然后再去使用普通攻击。使用冲撞撞击的目标一定是选择生命最高的怪物，同样也可以用一个优先队列维护所有的怪物的生命值，每次取出最大的生命值进行打击即可。若技能使用光了就只能使用普通攻击。

```

1  #include <bits/stdc++.h>
2
3  const int maxn = 2e5 + 10;
4  typedef long long ll;
5  using namespace std;
6
7  int n, t, w;
8  int cas, tol, T;
9  ll a[maxn];
10
11 bool check(int x) {
12     multiset<int> s;
13     for (int i = 1; i <= n; i++) s.insert(a[i]);
14     int bi = w, tim = t;
15     while (tim--) {
16         int res = *--s.end();
17         s.erase(--s.end());
18         if (bi) {
19             if (res > x) s.insert(res - x);
20             bi--;
21         }
22         if (s.size() == 0) break;
23         if (bi == 0) break;
24     }
25     ll tmp = 0;
26     for (auto it : s) tmp += 1ll * it;
27     if (tmp > 1ll * tim) return 0;
28     return 1;
29 }
30
31 int main() {
32     scanf ("%d", &T);
33     while (T --) {
34         scanf ("%d %d %d", &n, &t, &w);
35         ll res = 0;
36         for (int i = 1; i <= n; i++) {
37             scanf ("%lld", &a[i]), res += a[i];
38         }
39         int l = 2, r = 1e9, ans = -1;
40         if (res <= t) {
41             puts("0");
42             return 0;
43         }
44         while (l <= r) {
45             int mid = l + r >> 1;
46             if (check(mid)) ans = mid, r = mid - 1;
47             else l = mid + 1;
48         }
49         printf("%d\n", ans);
50     }
51     return 0;
52 }

```

G: 小胖修路

动态规划，用 $dp[i][j][0/1]$ 表示前 i 条路制造 j 个坎且第 i 条路不当 (0) 或当 (1) 坎时的最小花费。

显然，第 i 条路变成了坎，那么第 $i - 1, i + 1$ 条路都不可能是坎，所以直接从 $i - 2$ 的状态来转移，讨论 $i - 2$ 是不是坎即可。

第 i 条路不是坎，那么直接从状态 $i - 1$ 转移就好了，讨论 $i - 1$ 是不是坎即可。

转移方程：

$$dp[i][j][0] = \min(dp[i-1][j][0], dp[i-1][j][1] + \max(a[i] - a[i-1] + 1, 0))$$

$$dp[i][j][1] = \min(dp[i-2][j-1][1] + \max(a[i-1] - \min(a[i-2], a[i]) + 1, 0), dp[i-2][j-1][0] + \max(a[i-1] - a[i]$$

然后注意一下 dp 方程的初始化和一些边界即可，最后答案就是 $\min(dp[n][k][0], dp[n][k][1])$ 。

代码1:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int N = 5e3 + 5, M = 1e6, P = 1e9 + 7, INF = 0x3f3f3f3f;
5
6 int n, k;
7 int a[N], f[N][N][2];
8
9 int main() {
10     ios::sync_with_stdio(false);
11     cin.tie(0);
12     int T;
13     cin >> T;
14     while (T--) {
15         cin >> n >> k;
16         for (int i = 1; i <= n; i++) {
17             cin >> a[i];
18         }
19         for (int i = 0; i <= n; i++) for (int j = 0; j <= n; j++) f[i][j][0] = f[i][j][1] = INF;
20         if (k > (n + 1) / 2) {
21             cout << -1 << '\n';
22             continue;
23         }
24         a[0] = INF, f[0][0][0] = f[1][0][0] = f[1][1][1] = 0;
25         for (int i = 2; i <= n; i++) {
26             f[i][0][0] = 0;
27             for (int j = 1; j <= i; j++) {
28                 f[i][j][0] = min({f[i][j][0], f[i-1][j][0], f[i-1][j][1] + max(0, a[i] - a[i-1] + 1)}),
29                 f[i][j][1] = min({f[i][j][1], f[i-2][j-1][1] + max(0, a[i-1] - min(a[i-2], a[i]) + 1),
30                 f[i-2][j-1][0] + max(a[i-1] - a[i] + 1, 0)});
31             }
32             cout << min(f[n][k][0], f[n][k][1]) << '\n';
33         }
34         return 0;
35     }
36 }
```

H:小胖旅行

树上差分模板题，对于每组 u, v ，在 $a[u] ++, a[v] ++, a[Lca(u, v)] --, a[fa[Lca(u, v)]] --$ ，从下往上差分即可。

$Lca(u, v)$ 为 u, v 的最近公共祖先。

代码1:

倍增求 lca

```
1 #include <bits/stdc++.h>
2
3 #define pb push_back
4
5 using namespace std;
6
7 typedef long long ll;
8 const int maxn = 1e6 + 100;
9
10 int n, m;
11 int a[maxn], fa[maxn][30], dep[maxn], b[maxn], c[maxn];
12 vector<int> e[maxn];
13 void dfs(int u, int f, int d) {
14     dep[u] = d;
```

```

15     for(auto v : e[u]) {
16         if(v == f) continue;
17         fa[v][0] = u;
18         dfs(v, u, d+1);
19     }
20 }
21 void init() {
22     for(int j = 1; j <= 20; j++) {
23         for(int i = 1; i <= n; i++) {
24             fa[i][j] = fa[fa[i][j-1]][j-1];
25         }
26     }
27 }
28 int LCA(int u, int v) {
29     if(dep[u] < dep[v]) swap(u, v);
30     int f = dep[u] - dep[v];
31     for(int i = 0; (1<<i) <= f; i++) {
32         if((1<<i) & f) u = fa[u][i];
33     }
34     if(u == v) return u;
35     for(int i = 20; i >= 0; i--) {
36         if(fa[u][i] != fa[v][i]) {
37             u = fa[u][i];
38             v = fa[v][i];
39         }
40     }
41     return fa[u][0];
42 }
43 void dfs1(int u, int f) {
44     for(auto v : e[u]) {
45         if(v == f) continue;
46         dfs1(v, u);
47         b[u] += b[v];
48     }
49 }
50 int main() {
51     scanf("%d", &n, &m);
52     for(int i = 1; i <= m; i++) scanf("%d", &a[i]);
53     a[m+1] = a[1];
54     for(int i = 1, x, y; i < n; i++) {
55         scanf("%d%d", &x, &y);
56         e[x].pb(y); e[y].pb(x);
57     }
58     dfs(1, 1, 0);
59     init();
60     for(int i = 2, f; i <= m+1; i++) {
61         f = LCA(a[i-1], a[i]);
62         b[fa[f][0]]--; b[a[i]]++; b[a[i-1]]++;
63         b[f]--;
64     }
65     dfs1(1, 1);
66     for(int i = 2; i <= m; i++) b[a[i]]--;
67     for(int i = 1; i <= n; i++) printf("%d\n", b[i]);
68 }
69

```

代码2:

树剖求 lca

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4
5  const int N = 1e5 + 5;
6
7  int n, m, a[N], b[N];
8  int sz[N], fa[N], son[N], dep[N], topf[N];
9  std::vector<int> g[N];
10
11 void dfs1(int u, int f) {
12     sz[u] = 1, fa[u] = f, dep[u] = dep[f] + 1;
13     for (auto v : g[u]) {
14         if (v == f) continue;
15         dfs1(v, u);
16         sz[u] += sz[v];
17         if (!son[u] || sz[son[u]] < sz[v]) son[u] = v;
18     }
19 }
20
21 void dfs2(int u, int topf) {
22     topf[u] = topf;
23     if (son[u]) dfs2(son[u], topf);

```

```

24     for (auto v : g[u]) {
25         if (v == fa[u] || v == son[u]) continue;
26         dfs2(v, v);
27     }
28 }
29
30 int LCA(int u, int v) {
31     for (; top[u] ^ top[v]; u = fa[top[u]]) if (dep[top[u]] < dep[top[v]]) swap(u, v);
32     return dep[u] < dep[v] ? u : v;
33 }
34
35 void dfs(int u, int f) {
36     for (auto v : g[u]) {
37         if (v == f) continue;
38         dfs(v, u);
39         b[u] += b[v];
40     }
41 }
42
43 int main() {
44     ios::sync_with_stdio(false);
45     cin.tie(nullptr);
46     cin >> n >> m;
47     for (int i = 1; i <= m; i++) {
48         cin >> a[i];
49     }
50     for (int i = 2, u, v; i <= n; i++) {
51         cin >> u >> v;
52         g[u].push_back(v), g[v].push_back(u);
53     }
54     dfs1(1, 0), dfs2(1, 0);
55     a[m + 1] = a[1];
56     for (int i = 2; i <= m + 1; i++) {
57         int u = a[i - 1], v = a[i], lca = LCA(u, v);
58         b[u]++, b[v]++, b[lca]--, b[fa[lca]]--;
59     }
60     dfs(1, 0);
61     for (int i = 2; i <= m; i++) b[a[i]]--;
62     for (int i = 1; i <= n; i++) cout << b[i] << '\n';
63     return 0;
64 }

```