다이내믹 프로그래밍 4 (연습)

소스 코드

최백준 choi@startlink.io

C++14

```cpp
#include <cstdio>
#include <cstring>
int a[16][16];
char str[20];
int d[1<<16];
int main() {
    int n;
    scanf("%d",&n);
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            scanf("%d",&a[i][j]);
        }
    }
    scanf("%s",str);
    int start = 0;
    memset(d,-1,sizeof(d));
    for (int i=n-1; i>=0; i--) {
        start = start * 2;
        if (str[i] == 'Y') {
            start += 1;
        }
    }
    int p;
    scanf("%d",&p);
    d[start] = 0;
    for (int i=0; i<(1<<n); i++) {
        if (d[i] == -1) continue;
        for (int j=0; j<n; j++) {
            if (i&(1<<j)) { // j가 켜져 있음
                for (int k=0; k<n; k++) {
                    if ((i&(1<<k))==0) { // k가 꺼져있음
                        if (d[i|(1<<k)] == -1 || d[i|(1<<k)] > d[i] + a[j][k]) {
                            d[i|(1<<k)] = d[i]+a[j][k];
                        }
                    }
                }
            }
        }
    }
    int ans = -1;
    for (int i=0; i<(1<<n); i++) {
        if (d[i] == -1) continue;
        int cnt = 0;
        for (int j=0; j<n; j++) {
            if (i&(1<<j)) {
                cnt += 1;
            }
        }
        if (cnt >= p) {
            if (ans == -1 || ans > d[i]) {
                ans = d[i];
            }
        }
    }
    printf("%d\n",ans);

    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	1372 KB	16 ms	1359 B

C++

```cpp
#include <iostream>
#include <iomanip>
using namespace std;
double a[20][20];
bool c[1<<20];
double d[1<<20];
int n;
double go(int i, int s) {
    if (i == n) {
        return 1.0;
    }
    if (c[s]) {
        return d[s];
    }
    c[s] = true;
    double &ans = d[s];
    for (int k=0; k<n; k++) {
        if ((s&(1<<k)) == 0) {
            double temp = a[i][k]*go(i+1, s|(1<<k));
            if (ans < temp) {
                ans = temp;
            }
        }
    }
    return ans;
}
int main() {
    cin >> n;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            a[i][j] /= 100.0;
        }
    }
    cout << fixed << setprecision(10) << go(0, 0)*100.0 << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	11208 KB	168 ms	737 B

C++14

```cpp
#include <iostream>
#include <cstring>
#include <vector>
#include <algorithm>
using namespace std;
int a[1001][3];
long long d[1001][1001][1<<3];
const long long inf = 1000000000000000000LL;
int states[11][3] = {
    {0, 0, 0},
    {1, 1, 1},
    {2, 2, 1},
    {4, 4, 1},
    {0, 3, 1},
    {0, 6, 1},
    {1, 7, 2},
    {4, 7, 2},
    {3, 3, 2},
    {6, 6, 2},
    {5, 5, 2}
};
long long calc(int n, int cur) {
    long long ans = 0;
    if (n < 0 && cur != 0) return -inf;
    if (cur & 1) ans += a[n][0];
    if (cur & 2) ans += a[n][1];
    if (cur & 4) ans += a[n][2];
    return ans;
}
long long go(int n, int k, int state) {
    if (n < 0) {
        if (k == 0) return 0;
        else return -inf;
    }
    if (k == 0) return 0;
    if (k < 0) return -inf;
    long long &ans = d[n][k][state];
    if (ans != -inf) return ans;
    for (int i=0; i<11; i++) {
        int prev = states[i][0];
        int cur = states[i][1];
        int cost = states[i][2];
        if (cur & state) continue;
        ans = max(ans, calc(n, cur) + go(n-1, k-cost, prev));
    }
    return ans;
}
int main() {
    int n, m, k;
    cin >> n >> k;
    m = 3;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    fill(&d[0][0][0], &d[1001][0][0], -inf);
    cout << go(n-1, k, 0) << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	64624 KB	312 ms	1367 B

C++14

```cpp
#include <iostream>
#include <vector>
#include <string>
using namespace std;
long long d[1<<15][100];
long long gcd(long long x, long long y) {
    if (y == 0) return x;
    else return gcd(y, x%y);
}
int main() {
    int n;
    cin >> n;
    vector<string> num(n);
    vector<int> a(n);
    vector<int> len(n);
    for (int i=0; i<n; i++) {
        cin >> num[i];
        len[i] = num[i].size();
    }
    int k;
    cin >> k;
    for (int i=0; i<n; i++) {
        for (int j=0; j<len[i]; j++) {
            a[i] = a[i] * 10 + (num[i][j] - '0');
            a[i] %= k;
        }
    }
    vector<int> ten(51);
    ten[0] = 1;
    for (int i=1; i<=50; i++) {
        ten[i] = ten[i-1] * 10;
        ten[i] %= k;
    }
    d[0][0] = 1;
    for (int i=0; i<(1<<n); i++) {
        for (int j=0; j<k; j++) {
            for (int l=0; l<n; l++) {
                if ((i&(1<<l)) == 0) {
                    int next = j * ten[len[l]];
                    next %= k;
                    next += a[l];
                    next %= k;
                    d[i|(1<<l)][next] += d[i][j];
                }
            }
        }
    }
    long long t1 = d[(1<<n)-1][0];
    long long t2 = 1;
    for (int i=2; i<=n; i++) {
        t2 *= (long long)i;
    }
    long long g = gcd(t1,t2);
    t1 /= g;
    t2 /= g;
    cout << t1 << '/' << t2 << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	27592 KB	252 ms	1353 B

C++14

```cpp
#include <iostream>
#include <vector>
#include <set>
using namespace std;
set<int> d[1<<6];
bool is_prime(int x) {
    if (x < 2) {
        return false;
    }
    for (int i=2; i*i <= x; i++) {
        if (x % i == 0) {
            return false;
        }
    }
    return true;
}
set<int> solve(vector<int> &a, int mask) {
    set<int> &ans = d[mask];
    if (ans.size() > 0) {
        return ans;
    }
    if (mask == 0) {
        return ans;
    }
    int n = a.size();
    for (int i=0; i<(1<<n); i++) {
        int first = i & mask;
        int second = ~first & mask;
        if (first == 0 || second == 0) {
            continue;
        }
        solve(a, first);
        solve(a, second);

        for (int x : d[first]) {
            for (int y : d[second]) {
                if (x != 0) {
                    ans.insert(y/x);
                }
                if (y != 0) {
                    ans.insert(x/y);
                }
                ans.insert(x+y);
                ans.insert(x-y);
                ans.insert(y-x);
                ans.insert(x*y);
            }
        }
    }
    return ans;
}
int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
        d[1<<i].insert(a[i]);
    }

    auto ans = solve(a, (1<<n)-1);

    int ans_min=-1;
    int ans_max=-1;

    for (auto &x : ans) {
        if (is_prime(x)) {
            if (ans_min == -1) {
                ans_min = x;
            }
            ans_max = x;
        }
    }

    if (ans_min == -1) {
        cout << -1 << '\n';
    } else {
        cout << ans_min << '\n';
        cout << ans_max << '\n';
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	6484 KB	140 ms	1686 B

C++14

```cpp
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <cstring>
using namespace std;
pair<int,int> d[2501][51][1<<4];
string a[4];
string str;
pair<int,int> go(int index, int covered, int mask) {
    if (mask == 0) {
        return make_pair(0, 0);
    }
    auto &ans = d[index][covered][mask];
    if (ans.first != -1) {
        return ans;
    }
    ans = make_pair(1000000, -1000000);
    if (index == str.size()) {
        return ans;
    }
    ans = go(index+1, max(0, covered-1), mask);
    for (int i=0; i<4; i++) {
        if (mask & (1<<i)) {
            if (str.substr(index, a[i].size()) == a[i]) {
                int nextc = 0;
                if (a[i].size() > covered) {
                    nextc = (int)a[i].size() - covered;
                }
                auto temp = go(index, max(covered, (int)a[i].size()), mask - (1<<i));
                ans = make_pair(min(temp.first+nextc, ans.first), max(temp.second+nextc, ans.second));
            }
        }
    }
    return ans;
}
int main() {
    cin >> str;
    for (int i=0; i<4; i++) {
        cin >> a[i];
    }
    sort(a,a+4);
    memset(d,-1,sizeof(d));
    auto ans = go(0, 0, 15);
    cout << ans.first << ' ' << ans.second << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	18208 KB	244 ms	1258 B

C++14

```cpp
#include <iostream>
#include <cstring>
using namespace std;
char a[20][20];
int d[14*14][1<<14];
int n,m;
int score[6][6] = {
    {10,8,7,5,0,1},
    {8,6,4,3,0,1},
    {7,4,3,2,0,1},
    {5,3,2,2,0,1},
    {0,0,0,0,0,0},
    {1,1,1,1,0,0}
};
int calc(char x, char y) {
    return score[x-'A'][y-'A'];
}
int go(int num, int state) {
    if (num >= n*m) {
        return 0;
    }
    if (d[num][state] >= 0) {
        return d[num][state];
    }
    int &ans = d[num][state];
    ans = 0;
    int x = num/m;
    int y = num%m;
    ans = go(num+1, (state >> 1));
    if ((state & 1) == 0) {
        if (x != n-1) {
            int temp = go(num+1, (state >> 1) | (1 << (m-1)));
            temp += calc(a[x][y], a[x+1][y]);
            ans = max(ans, temp);
        }
        if (y != m-1 && (state & 2) == 0) {
            int temp = go(num+2, (state >> 2));
            temp += calc(a[x][y], a[x][y+1]);
            ans = max(ans, temp);
        }
    }
    return ans;
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    memset(d,-1,sizeof(d));
    cout << go(0, 0) << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	14532 KB	44 ms	1134 B

C++14

```cpp
#include <iostream>
#include <vector>
using namespace std;
const long long mod = 1000000007;
using matrix = vector<vector<long long>>;
matrix operator*(const matrix &a, const matrix &b) {
    int n = a.size();
    matrix c(n, vector<long long>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            for (int k=0; k<n; k++) {
                c[i][j] += a[i][k] * b[k][j];
                c[i][j] %= mod;
            }
        }
    }
    return c;
}
int main() {
    int n, m;
    cin >> n >> m;
    matrix ans(m, vector<long long>(m));
    matrix even(m, vector<long long>(m));
    matrix odd(m, vector<long long>(m));
    matrix a(m, vector<long long>(m));
    for (int i=0; i<m; i++) {
        ans[i][i] = 1;
        odd[i][i] = 1;
        if (i-1 >= 0) {
            odd[i][i-1] = 1;
            even[i][i-1] = 1;
        }
        if (i+1 < m) {
            odd[i][i+1] = 1;
            even[i][i+1] = 1;
        }
    }
    n -= 1;
    a = even * odd;
    int p = n/2;
    while (p > 0) {
        if (p % 2 == 1) {
            ans = ans * a;
        }
        a = a * a;
        p /= 2;
    }
    if (n % 2 == 1) {
        ans = ans * even;
    }
    long long sum = 0;
    for (int i=0; i<m; i++) {
        for (int j=0; j<m; j++) {
            sum += ans[i][j];
            sum %= mod;
        }
    }
    cout << sum << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	1984 KB	4 ms	1378 B

C++14

```cpp
#include <iostream>
#include <vector>
using namespace std;
const long long mod = 1000000007;
using matrix = vector<vector<long long>>;
matrix operator * (const matrix &a, const matrix &b) {
    int n = a.size();
    matrix c(n, vector<long long>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            for (int k=0; k<n; k++) {
                c[i][j] += a[i][k] * b[k][j];
                c[i][j] %= mod;
            }
        }
    }
    return c;
}
int state_map[10][2][2];
int main() {
    long long n;
    int k;
    cin >> n >> k;
    int kk = max(k, 1);
    int state_num = 0;
    for (int a=0; a<kk; a++) {
        for (int b=0; b<2; b++) {
            for (int c=0; c<2; c++) {
                state_map[a][b][c] = state_num++;
            }
        }
    }
    int size = kk * 2 * 2;
    matrix ans(size, vector<long long>(size));
    matrix mat(size, vector<long long>(size));
    for (int i=0; i<size; i++) {
        ans[i][i] = 1;
    }
    for (int a=0; a<kk; a++) {
        for (int b=0; b<2; b++) {
            for (int c=0; c<2; c++) {
                int state = state_map[a][b][c];
                if (k > 0) {
                    mat[state][state_map[(a+1)%k][0][c]] += 1;
                }
                if (b == 0) {
                    mat[state][state_map[a][1][c]] += 1;
                }
                mat[state][state_map[a][0][1]] += 1;
            }
        }
    }
    while (n > 0) {
        if (n % 2 == 1) {
            ans = ans * mat;
        }
        mat = mat * mat;
        n /= 2;
    }
    long long sum = ans[0][state_map[0][0][1]] + ans[0][state_map[0][1][1]];
    sum %= mod;
    cout << sum << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	1984 KB	20 ms	1690 B

C++14

```cpp
#include <iostream>
#include <vector>
using namespace std;
const long long mod = 1000000007;
using matrix = vector<vector<long long>>;
matrix operator*(const matrix &a, const matrix &b) {
    int n = a.size();
    matrix c(n, vector<long long>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            for (int k=0; k<n; k++) {
                c[i][j] += a[i][k] * b[k][j];
                c[i][j] %= mod;
            }
        }
    }
    return c;
}
int isset(int state, int index) {
    if (state & (1 << index)) {
        return 1;
    } else {
        return 0;
    }
}
bool ok(int pstate, int state, int m) {
    for (int i=0; i<m-1; i++) {
        int color11 = isset(pstate, i);
        int color12 = isset(pstate, i+1);
        int color21 = isset(state, i);
        int color22 = isset(state, i+1);
        if (color11 == color12 && color12 == color21 && color21 == color22) {
            return false;
        }
    }
    return true;
}
int main() {
    long long n;
    int m;
    cin >> n >> m;
    int size = (1<<m);
    matrix ans(size, vector<long long>(size));
    matrix a(size, vector<long long>(size));
    for (int i=0; i<size; i++) {
        ans[i][i] = 1;
        for (int j=0; j<size; j++) {
            if (ok(i, j, m)) {
                a[i][j] = 1;
            }
        }
    }
    n -= 1;
    while (n > 0) {
        if (n % 2 == 1) {
            ans = ans * a;
        }
        a = a * a;
        n /= 2;
    }
    long long sum = 0;
    for (int i=0; i<size; i++) {
        for (int j=0; j<size; j++) {
            sum += ans[i][j];
            sum %= mod;
        }
    }
    cout << sum << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	1984 KB	12 ms	1671 B

C++14

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
typedef vector<vector<long long>> matrix;
const long long mod = 1000000009LL;
matrix operator * (const matrix &a, const matrix &b) {
    int n = a.size();
    matrix c(n, vector<long long>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            for (int k=0; k<n; k++) {
                c[i][j] += a[i][k] * b[k][j];
                c[i][j] %= mod;
            }
        }
    }
    return c;
}
int dx[] = {-2,-1,1,2};
int dy[] = {1,0,0,1};
matrix make(int n, int m) {
    auto on = [&](int state, int col, int row) {
        return state & (1 << (col*n+row));
    };
    auto possible = [&](int state, int next) {
        for (int i=0; i<n; i++) {
            if (!on(next, 0, i)) continue;
            for (int k=0; k<4; k++) {
                if (i+dx[k] < 0 || i+dx[k] >= n) continue;
                if (on(state, dy[k], i+dx[k])) return false;
            }
        }
        return true;
    };
    int states = (1 << 2*n);
    matrix adj = vector<vector<long long>>(states, vector<long long>(states));
    for (int i=0; i<states; i++) {
        for (int j=0; j<(1<<n); j++) {
            if (possible(i, j)) {
                int k = (i >> n) | (j << n);
                adj[i][k] = 1;
            }
        }
    }
    return adj;
}
matrix pow(matrix a, int m) {
    int n = a.size();
    matrix ans = vector<vector<long long>>(n, vector<long long>(n));
    for (int i=0; i<n; i++) {
        ans[i][i] = 1;
    }
    while (m > 0) {
        if (m % 2 == 1) {
            ans = ans * a;
        }
        a = a * a;
        m /= 2;
    }
    return ans;
}
int main() {
    int t;
    cin >> t;
    while (t--) {
        int n, m;
        cin >> n >> m;
        matrix adj = make(n, m);
        matrix ans = pow(adj, m);
        long long sum = 0;
        for (int i=0; i<(1<<(2*n)); i++) {
            sum = (sum + ans[0][i]) % mod;
        }
        cout << sum << '\n';
    }
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	3980 KB	6052 ms	2007 B

끝

코드 플러스

https://code.plus

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.