

다이나믹 프로그래밍 4

소스코드

최백준 choi@startlink.io

C++14

```
1 #include <cstdio>
2 long long d[101][10][1<<10];
3 long long mod = 1000000000LL;
4 int main() {
5     int n;
6     scanf("%d",&n);
7     for (int i=1; i<=9; i++) {
8         d[1][i][1<<i] = 1LL;
9     }
10    for (int i=1; i<=n-1; i++) {
11        for (int j=0; j<=9; j++) {
12            for (int k=0; k<(1<<10); k++) {
13                if (d[i][j][k] == 0) {
14                    continue;
15                }
16                if ((k > (1<<j)) == 0) {
17                    continue;
18                }
19                if (j+1 <= 9) {
20                    d[i+1][j+1][k | (1<<(j+1))] += d[i][j][k];
21                    d[i+1][j+1][k | (1<<(j+1))] %= mod;
22                }
23                if (j-1 >= 0) {
24                    d[i+1][j-1][k | (1<<(j-1))] += d[i][j][k];
25                    d[i+1][j-1][k | (1<<(j-1))] %= mod;
26                }
27            }
28        }
29    }
30
31    long long ans = 0;
32    for (int i=0; i<=9; i++) {
33        ans += d[n][i][(1<<10)-1];
34        ans %= mod;
35    }
36    printf("%lld\n",ans);
37    return 0;
38 }
```

Handwritten notes and diagrams:

- Diagram showing the transition from state i to j with a value of 1, and a note $2 \rightarrow 3$.
- Diagram showing the transition from state i to $j+1$ with a value of 1, and a note $2 \rightarrow 3$.
- Diagram showing the transition from state i to $j-1$ with a value of 1, and a note $2 \rightarrow 3$.
- Diagram showing the transition from state i to j with a value of 1, and a note $2 \rightarrow 3$.
- Diagram showing the transition from state i to j with a value of 1, and a note $2 \rightarrow 3$.

C++14

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 const long long mod = 1000000000;
5 long long d[101][10][10][10];
6 // D[len][last][left][right]
7 // length: len, last digit: last
8 // left ~ right
9 int n;
10 long long go(int len, int last, int left, int right) {
11     if (len == n) {
12         if (left == 0 && right == 9) {
13             return 1;
14         } else {
15             return 0;
16         }
17     }
18     long long &ans = d[len][last][left][right];
19     if (ans != -1) return ans;
20     ans = 0;
21     // last + 1
22     if (last + 1 <= 9) {
23         if (last+1 <= right) {
24             ans += go(len+1, last+1, left, right);
25         } else {
26             ans += go(len+1, last+1, left, last+1);
27         }
28     }
29     // last - 1
30     if (last - 1 >= 0) {
31         if (last-1 >= left) {
32             ans += go(len+1, last-1, left, right);
33         } else {
34             ans += go(len+1, last-1, last-1, right);
35         }
36     }
37     ans %= mod;
38     return ans;
39 }
40 int main() {
41     cin >> n;
42     memset(d, -1, sizeof(d));
43     long long ans = 0;
44     for (int i=1; i<=9; i++) {
45         ans += go(1, i, i, i);
46     }
47     ans %= mod;
48     cout << ans << '\n';
49     return 0;
50 }
```

3, 1 0/2400 [1, 1]

last -> last+1

1/2 1/2 last right

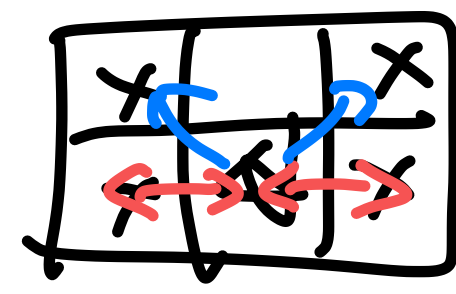
[1, 1]

C++14

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int n, m;
5 char a[22][22];
6 int d[10][1<<10];
7 bool isset(int state, int i) {
8     return (state & (1 << i)) > 0;
9 }
10 bool check(int row, int state) {
11     if (row < 0) return true;
12     for (int j=0; j<m-1; j++) {
13         if (isset(state, j) && isset(state, j+1)) {
14             return false;
15         }
16     }
17     for (int j=0; j<m; j++) {
18         if (a[row][j] == 'x' && isset(state, j)) {
19             return false;
20         }
21     }
22     return true;
23 }
24 int main() {
25     int t;
26     cin >> t;
27     while (t--) {
28         cin >> n >> m;
29         for (int i=0; i<n; i++) {
30             cin >> a[i];
31         }
32         memset(d, 0, sizeof(d));
33         int ans = 0;
34         for (int row=0; row<n; row++) {
35             for (int state=0; state<(1<<m); state++) {
36                 if (!check(row, state)) continue;
37                 // row, state
38                 // row-1, pstate
39                 for (int pstate=0; pstate<(1<<m); pstate++) {
40                     if (!check(row-1, pstate)) continue;
41                     int cnt = 0;
42                     bool ok = true;
43                     for (int j=0; j<m; j++) {
44                         if (isset(state, j)) {
45                             cnt += 1;
46                             if (j-1 >= 0 && isset(pstate, j-1)) ok = false;
47                             if (j+1 < m && isset(pstate, j+1)) ok = false;
48                         }
49                     }
50                     if (ok) {
51                         if (row == 0) {
52                             d[row][state] = max(d[row][state], cnt);
53                         } else {
54                             d[row][state] = max(d[row][state], d[row-1][pstate] + cnt);
55                         }
56                     }
57                 }
58             }
59         }
60         for (int state=0; state<(1<<m); state++) {
61             if (ans < d[n-1][state]) {
62                 ans = d[n-1][state];
63             }
64         }
65         cout << ans << '\n';
66     }
67     return 0;
68 }
```

32비트
DP[10][1024] = 컨닝의 상태
32비트 저장

상태 state
[배열]



row 행이 상태 state 가능

1) 32비트의 범위 0~1023

2) <-> 1

row 행

state

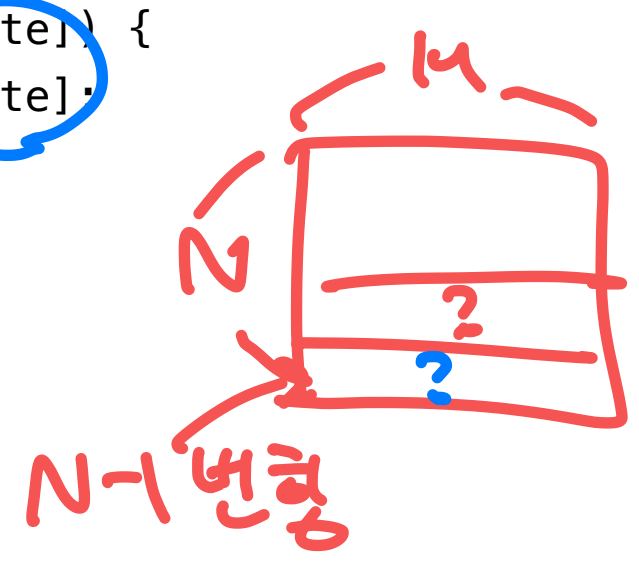
모든 상태

row-1

pstate

row-1의 j와 j+1

row-1 행
row



$(N \times 2^M) \times 2^M$
이전 행의 상태

$= N \times 2^{2M}$

$N, M \leq 10$

결과

메모리

시간

코드 길이

맞았습니다!!

2028 KB

60 ms

1978 B

$10 \times 2^{20} = 10485760$

C++14

```

1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int d[14*14][1<<14];
5 int n,m;
6 int go(int num, int state) {
7     if (num == n*m && state == 0) {
8         return 1;
9     }
10    if (num >= n*m) {
11        return 0;
12    }
13    if (d[num][state] >= 0) {
14        return d[num][state];
15    }
16    int &ans = d[num][state];
17    ans = 0;
18    if (state & 1) {
19        ans = go(num+1, (state >> 1));
20    } else {
21        ans = go(num+1, (state >> 1) | (1 << (m-1)));
22        if ((num%m) != (m-1) && (state & 2) == 0) {
23            ans += go(num+2, (state >> 2));
24        }
25    }
26    ans %= 9901;
27    return ans;
28 }
29 int main() {
30     cin >> n >> m;
31     memset(d, -1, sizeof(d));
32     cout << go(0, 0) << '\n';
33     return 0;
34 }

```

결과

메모리

시간

코드 길이

맞았습니다!!

14532 KB

16 ms

713 B

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <algorithm>
4 #include <vector>
5 #include <string>
6 using namespace std;
7 const int inf = 100000000;
8 int d[10*25][1<<10];
9 string a[25];
10 int n, m;
11 int go(int index, int state) {
12     if (index == n*m) {
13         if (state == 0) return 0;
14         else return -inf;
15     }
16     if (index > n*m) return -inf;
17     int &ans = d[index][state];
18     if (ans != -1) return ans;
19     int i = index/m;
20     int j = index%m;
21     if (a[i][j] == '1') {
22         ans = go(index+1, state >> 1) + 1;
23         return ans;
24     }
25     ans = go(index+1, state>>1);
26     if ((state & 1) == 0) {
27         ans = max(ans, go(index+1, state>>1) + 1);
28     }
29     if (i != n-1 && j != m-1 && (state & 1) == 0 && (state & 2) == 0) {
30         if (a[i][j+1] == '.' && a[i+1][j] == '.' && a[i+1][j+1] == '.') {
31             ans = max(ans, go(index+2, (state>>2) | (1<<(m-1)) | (1<<(m-2))) + 16);
32         }
33     }
34     return ans;
35 }
36 int main() {
37     cin >> n >> m;
38     vector<string> b(n);
39     for (int i=0; i<n; i++) {
40         cin >> b[i];
41     }
42     for (int j=0; j<m; j++) {
43         for (int i=0; i<n; i++) {
44             a[j] += b[i][j];
45         }
46     }
47     swap(n, m);
48     memset(d, -1, sizeof(d));
49     cout << go(0, 0) << '\n';
50     return 0;
51 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

2992 KB

0 ms

1238 B

C++14

```

1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <cstring>
5 using namespace std;
6 // 0 : -, 1: |
7 int d[11][1<<10];
8 int main() {
9     int n, m;
10    cin >> n >> m;
11    vector<string> a(n+1);
12    for (int i=1; i<=n; i++) {
13        cin >> a[i];
14    }
15    memset(d,-1,sizeof(d));
16    d[0][0] = 0;
17    for (int i=1; i<=n; i++) {
18        for (int prev=0; prev<(1<<m); prev++) {
19            if (d[i-1][prev] == -1) continue;
20            for (int cur=0; cur<(1<<m); cur++) {
21                int cnt = 0;
22                int last = -100;
23                // 0 : -, 1: |
24                for (int k=0; k<m; k++) {
25                    if (a[i][k] == '#') {
26                        continue;
27                    }
28                    if (cur & (1<<k)) {
29                        if (i-1==0) cnt++;
30                        else if ((prev & (1<<k)) == 0) cnt++;
31                        else if (a[i-1][k] == '#') cnt++;
32                    } else {
33                        if (last+1 != k) {
34                            cnt++;
35                        }
36                        last = k;
37                    }
38                }
39                if (d[i][cur] == -1 || d[i][cur] > d[i-1][prev]+cnt) {
40                    d[i][cur] = d[i-1][prev]+cnt;
41                }
42            }
43        }
44    }
45    int ans = -1;
46    for (int i=0; i<(1<<m); i++) {
47        if (d[n][i] == -1) continue;
48        if (ans == -1 || ans > d[n][i]) {
49            ans = d[n][i];
50        }
51    }
52    cout << ans << '\n';
53    return 0;
54 }

```

결과

메모리

시간

코드 길이

맞았습니다!!

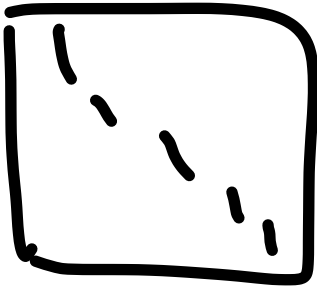
2032 KB

460 ms

1508 B

C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 typedef vector<vector<long long>> matrix;
5 const long long mod = 1000000007LL;
6 matrix operator * (const matrix &a, const matrix &b) {
7     int n = a.size();
8     matrix c(n, vector<long long>(n));
9     for (int i=0; i<n; i++) {
10         for (int j=0; j<n; j++) {
11             for (int k=0; k<n; k++) {
12                 c[i][j] += a[i][k] * b[k][j];
13             }
14             c[i][j] %= mod;
15         }
16     }
17     return c;
18 }
19 int main() {
20     long long n;
21
22     cin >> n;
23
24     if (n%2 == 1) {
25         cout << 0 << '\n';
26         return 0;
27     }
28
29     matrix ans = vector<vector<long long>>(8, vector<long long>(8));
30     matrix a = vector<vector<long long>>(8, vector<long long>(8));
31
32     for (int i=0; i<8; i++) {
33         ans[i][i] = 1;
34     }
35
36     a[0][7] = 1;
37     a[1][6] = 1;
38     a[2][5] = 1;
39     a[4][3] = 1;
40     a[3][4] = a[3][7] = 1;
41     a[6][1] = a[6][7] = 1;
42     a[5][2] = 1;
43     a[7][0] = a[7][3] = a[7][6] = 1;
44
45     while (n > 0) {
46         if (n % 2 == 1) {
47             ans = ans * a;
48         }
49         a = a * a;
50         n /= 2;
51     }
52
53     long long sum = 0;
54     for (int i=0; i<7; i++) {
55         sum += ans[0][i];
56     }
57
58     sum %= mod;
59
60     cout << sum << '\n';
61
62     return 0;
63 }
```



결과

메모리

시간

코드 길이

맞았습니다!!

1988 KB

0 ms

1236 B

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <vector>
4 #include <string>
5 using namespace std;
6 void print(int x) {
7     if (x < 10) {
8         cout << 0;
9     }
10    cout << x << '\n';
11 }
12 int d[100][100];
13 vector<int> a;
14 int calc(int x, int y, int a0, int a1, int n) {
15     if (n == 0) return a0;
16     if (n == 1) return a1;
17     memset(d, -1, sizeof(d));
18     a.push_back(a0);
19     a.push_back(a1);
20     d[a0][a1] = 1;
21     for (int i=2; i<=n; i++) {
22         int a2 = x*a1 + y*a0;
23         a2 = a2 % 100;
24         if (d[a1][a2] != -1) {
25             int prefix = d[a1][a2];
26             int t = i - d[a1][a2];
27             return a[prefix+(n-prefix)%t];
28         }
29         a.push_back(a2);
30         d[a1][a2] = i;
31         a0 = a1;
32         a1 = a2;
33     }
34     return a1;
35 }
36 int main() {
37     int x, y, a0, a1, n;
38     cin >> x >> y >> a0 >> a1 >> n;
39     int ans = calc(x, y, a0, a1, n);
40     print(ans);
41     return 0;
42 }
43
```

$[A_{n-2}][A_{n-1}]$

$a[prefix+(n-prefix)\%t]$

결과

메모리

시간

코드 길이

맞았습니다!!

2028 KB

0 ms

894 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.