

다이나믹 프로그래밍 3

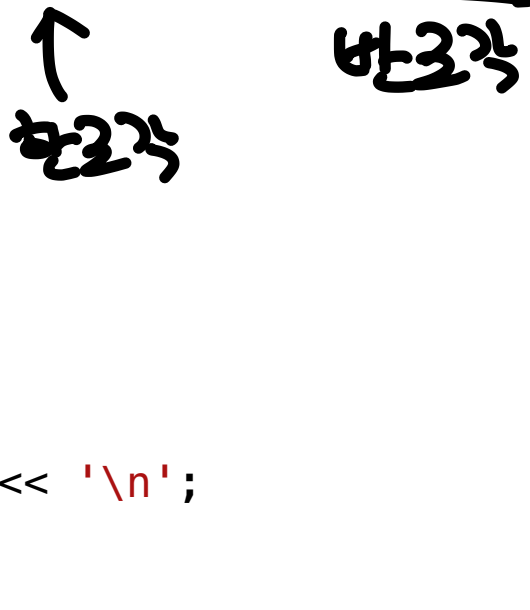
소스코드

최백준 choi@startlink.io



C++14

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 long long d[31][31];
5 long long calc(int f, int h) {
6     if (d[f][h] != -1) return d[f][h];
7     if (f == 0) return 1;
8     if (h == 0) return d[f][h] = calc(f-1, h+1);
9     return d[f][h] = calc(f-1, h+1) + calc(f, h-1);
10 }
11 int main() {
12     memset(d, -1, sizeof(d));
13     while (true) {
14         int n;
15         cin >> n;
16         if (n == 0) break;
17         cout << calc(n, 0) << '\n';
18     }
19     return 0;
20 }
```



결과	메모리	시간	코드 길이
맞았습니다!!	1992 KB	24 ms	455 B

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4 int n;
5 int a[500][500];
6 int d[500][500];
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 int go(int i, int j) {
10     if (d[i][j] != 0) {
11         return d[i][j];
12     }
13     d[i][j] = 1;
14     for (int k=0; k<4; k++) {
15         int x = i+dx[k];
16         int y = j+dy[k];
17         if (0 <= x && x < n && 0 <= y && y < n) {
18             if (a[i][j] < a[x][y]) {
19                 d[i][j] = max(d[i][j], go(x, y) + 1);
20             }
21         }
22     }
23     return d[i][j];
24 }
25
26 int main() {
27     cin >> n;
28     for (int i=0; i<n; i++) {
29         for (int j=0; j<n; j++) {
30             cin >> a[i][j];
31         }
32     }
33     int ans = 0;
34     for (int i=0; i<n; i++) {
35         for (int j=0; j<n; j++) {
36             ans = max(ans, go(i, j));
37         }
38     }
39     cout << ans << '\n';
40     return 0;
41 }
42
```

(i, j) 시작

.Memo

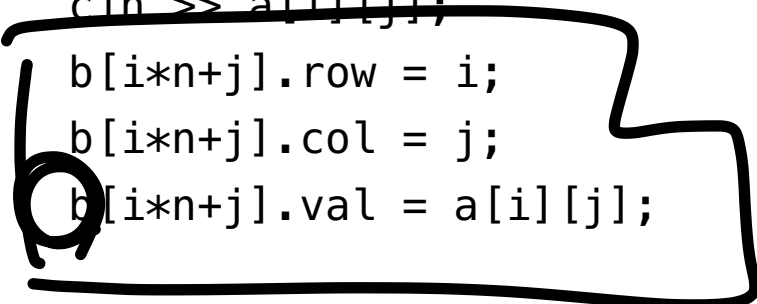
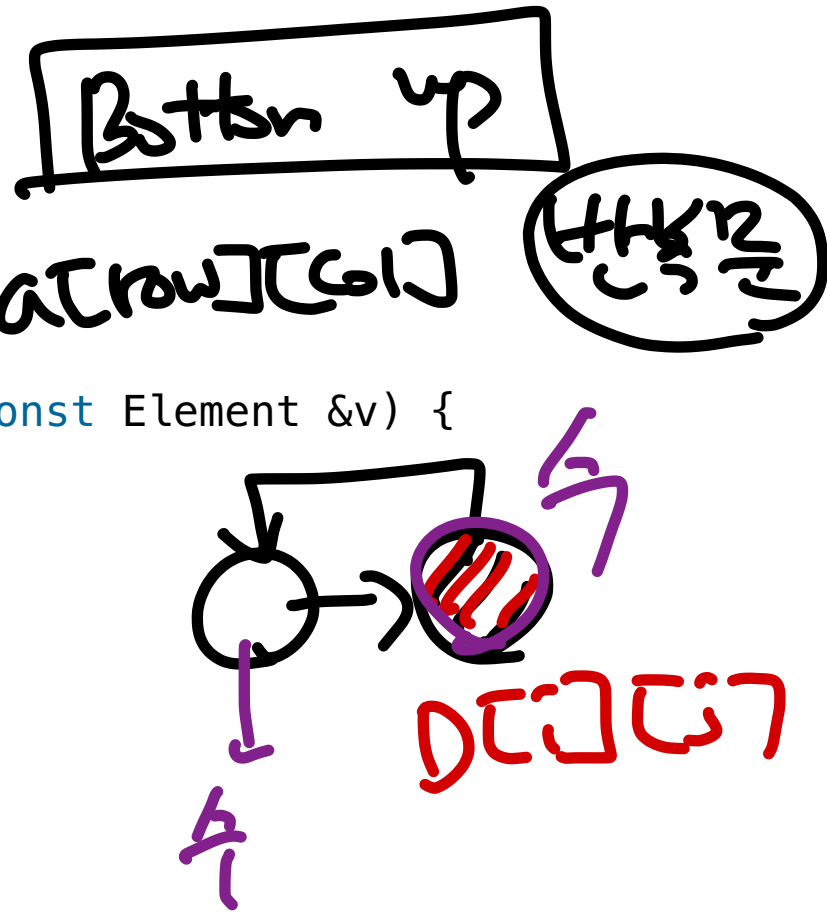
(i, j) → (x, y)

$O(N^2)$

(i, j)

C++14

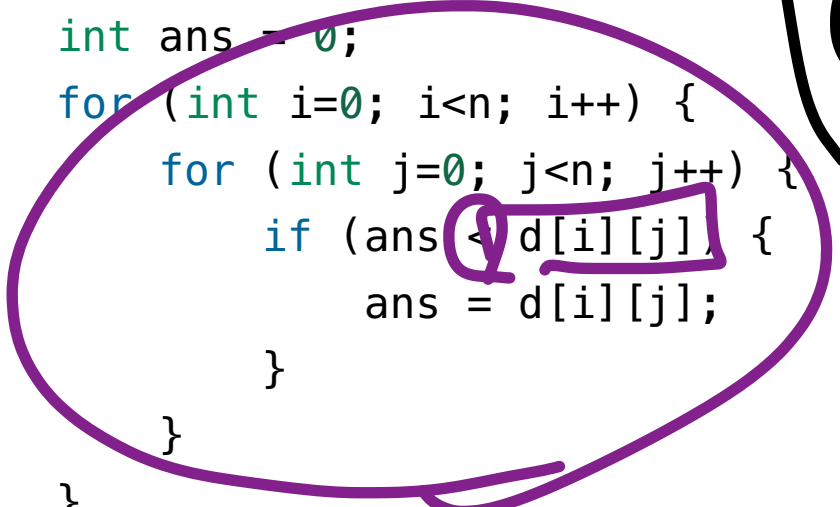
```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4 struct Element {
5     int row, col, val;
6 };
7 bool cmp(const Element &u, const Element &v) {
8     return u.val > v.val;
9 }
10 int a[500][500];
11 Element b[500*500];
12 int d[500][500];
13 int dx[] = {0,0,1,-1};
14 int dy[] = {1,-1,0,0};
15 int main() {
16     int n;
17     cin >> n;
18     for (int i=0; i<n; i++) {
19         for (int j=0; j<n; j++) {
20             cin >> a[i][j];
21             b[i*n+j].row = i;
22             b[i*n+j].col = j;
23             b[i*n+j].val = a[i][j];
24         }
25     }
26     sort(b, b+n*n, cmp);
27     for (int i=0; i<n*n; i++) {
28         int x = b[i].row;
29         int y = b[i].col;
30         d[x][y] = 1;
31         // (x, y) -> (nx, ny)
32         for (int k=0; k<4; k++) {
33             int nx = x+dx[k];
34             int ny = y+dy[k];
35             if (nx < 0 || nx >= n || ny < 0 || ny >= n) continue;
36             if (a[x][y] < a[nx][ny]) {
37                 d[x][y] = max(d[x][y], d[nx][ny]+1);
38             }
39         }
40     }
41     int ans = 0;
42     for (int i=0; i<n; i++) {
43         for (int j=0; j<n; j++) {
44             if (ans < d[i][j]) {
45                 ans = d[i][j];
46             }
47         }
48     }
49     cout << ans << '\n';
50     return 0;
51 }
```



크기(val) 내림차순

(x,y) → (nx,ny)

$O(N^2 \log N)$



C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 const long long mod = 1000000007LL;
5 long long d[11][100001];
6 long long sum[11];
7 vector<int> divisors[100001];
8 int main() {
9     int n, k;
10    cin >> n >> k;
11    for (int i=1; i<=k; i++) {
12        for (int j=i*2; j<=k; j+=i) {
13            divisors[j].push_back(i);
14        }
15    }
16    for (int j=1; j<=k; j++) {
17        d[1][j] = 1;
18    }
19    sum[1] = k;
20    for (int i=2; i<=n; i++) {
21        for (int j=1; j<=k; j++) {
22            d[i][j] = sum[i-1];
23            for (int div : divisors[j]) {
24                d[i][j] -= d[i-1][div];
25                d[i][j] %= mod;
26                d[i][j] += mod;
27                d[i][j] %= mod;
28            }
29            sum[i] += d[i][j];
30            sum[i] %= mod;
31        }
32    }
33    cout << sum[n] << '\n';
34    return 0;
35 }
```

$sum[i] = \sum d[i][j]$

$d[i][j] = \sum_{1 \leq c \leq k} (1 \leq c \leq k)$

$\tau + 1$

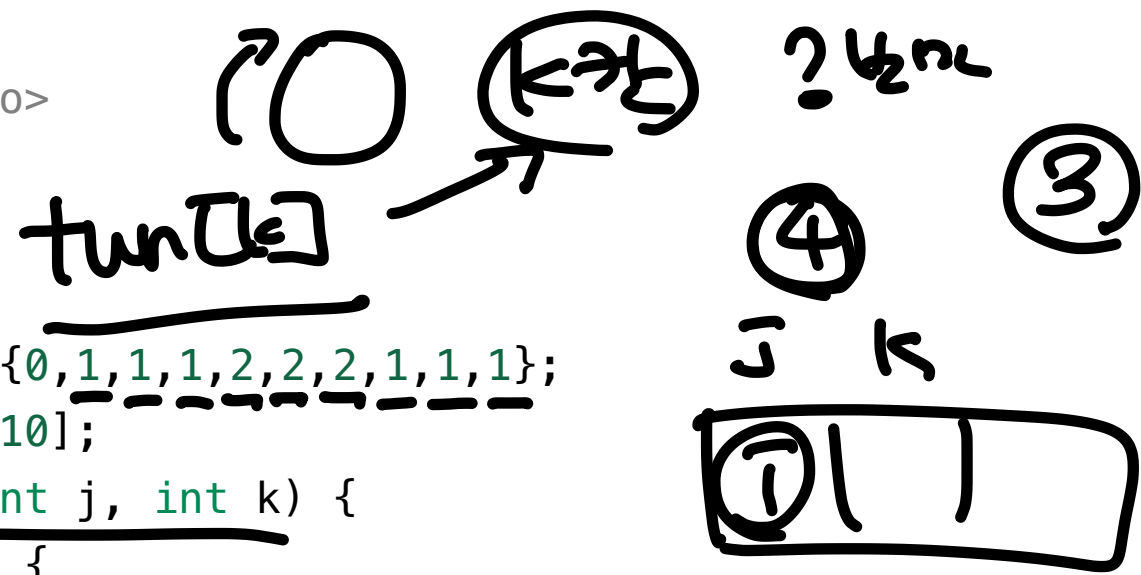
$O(N \cdot k \cdot \frac{100000}{k})$

$\rightarrow 10000$

결과	메모리	시간	코드 길이
맞았습니다!!	20976 KB	136 ms	819 B

C++14

```
1 #include <stdio>
2 int n;
3 int a[100];
4 int b[100];
5 int turn[10] = {0,1,1,1,2,2,2,1,1,1};
6 int d[100][10][10];
7 int go(int i, int j, int k) {
8     if (i == n) {
9         return 0;
10    }
11    if (d[i][j][k] != -1) {
12        return d[i][j][k];
13    }
14    int original = (a[i] + j) % 10;
15    int to = b[i];
16    for (int three = 0; three < 10; three++) {
17        for (int two = 0; two < 10; two++) {
18            int from = (original + two + three) % 10;
19            int one = to - from;
20            if (one < 0) one += 10;
21            int cost = turn[one] + turn[two] + turn[three];
22            cost += go(i+1, (k+two+three) % 10, three);
23            if (d[i][j][k] == -1 || d[i][j][k] > cost) {
24                d[i][j][k] = cost;
25            }
26        }
27    }
28    return d[i][j][k];
29 }
30 int main() {
31     scanf("%d",&n);
32     for (int i=0; i<n; i++) {
33         scanf("%1d",&a[i]);
34     }
35     for (int i=0; i<n; i++) {
36         scanf("%1d",&b[i]);
37     }
38     for (int i=0; i<n; i++) {
39         for (int j=0; j<10; j++) {
40             for (int k=0; k<10; k++) {
41                 d[i][j][k] = -1;
42             }
43         }
44     }
45     printf("%d\n",go(0,0,0));
46 }
```



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.