01背包优化（空间）：

```
#include<iostream>
#include<cstring>
using namespace std;
const int N = 1010;
int n, V, v[N], w[N];
int dp[N];

int main() {
    cin >> n >> V;
    for (int i = 1; i <= n; i++) {
        cin >> v[i] >> w[i];
    }
    for (int i = 1; i <= n; i++) {
        for (int j = V; j >= v[i]; j--) {
            dp[j] = max(dp[j], dp[j - v[i]] + w[i]);
        }
    }

cout << dp[V] << endl;


//2.
memset(dp, 0, sizeof dp);  // 将 dp 数组全部初始化为 0
for (int j = 1; j <= V; j++) dp[j] = -1; // 初始化第一行
for (int i = 1; i <= n; i++) {
    for (int j = V; j >= v[i]; j--) {
        if (dp[j - v[i]] != -1)
            dp[j] = max(dp[j], w[i] + dp[j - v[i]]);
    }
}

if (dp[V] == -1) {
    cout << 0 << endl; // 表示无法装入有效的物品组合
} else {
    cout << dp[V] << endl; // 输出最大价值
}

return 0;
}
```

BFS最短路径：433. 最小基因变化

```cpp
class Solution {
public:
    int minMutation(string startGene, string endGene, vector<string>& bank) {
        unordered_set<string> vis;                            // 判端在不在
        unordered_set<string> hash(bank.begin(), bank.end()); // 基因库

        string change = "ACGT";

        if (startGene == endGene)
            return 0;
        if (!hash.count(endGene))
            return -1;
        queue<string> q;
        q.push(startGene);
        vis.insert(startGene);
        int ret = 0;
        while (q.size()) {
            ret++;
            int sz = q.size();
            while (sz--) {
                string t = q.front();
                q.pop();
                for (int i = 0; i < 8; i++) {
                    string tmp = t;
                    for (int j = 0; j < 4; j++) {
                        tmp[i] = change[j];
                        if (hash.count(tmp) && !vis.count(tmp)) {
                            if(tmp == endGene)
                            return ret;
                            q.push(tmp);
                            vis.insert(tmp);
                        }
                    }
                }

            }
        }
        return -1;
    }
};
```