


Status	Time	Length	Lang	Submitted	Open	Share text 	RemoteRunId
Accepted	30ms	2337	C++11 5.3.0	2024-11-13 16:38:40			29960670

C++

```
1  #include <iostream>
2  using namespace std;
3
4  const int MAX_TORTUGAS = 5607;    // Máximo número de tortugas
5  const int PESO_MAXIMO = 10000000; // Peso muy alto para representar un límite exagerado
6
7  int peso[MAX_TORTUGAS];    // Arreglo para almacenar el peso de cada tortuga
8  int fuerza[MAX_TORTUGAS];  // Arreglo para almacenar la fuerza de cada tortuga
9  int dp[MAX_TORTUGAS + 1];  // Arreglo de DP para el peso mínimo de una pila con cierta altura
10
11 // Función para ordenar las tortugas por su fuerza usando un algoritmo de burbuja
12 void ordenarTortugas(int n)
13 {
14     for (int i = 0; i < n - 1; ++i)
15     {
16         for (int j = 0; j < n - i - 1; ++j)
17         {
18             if (fuerza[j] > fuerza[j + 1])
19             {
20                 // Intercambiar peso de las tortugas
21                 int tempPeso = peso[j];
22                 peso[j] = peso[j + 1];
23                 peso[j + 1] = tempPeso;
24
25                 // Intercambiar fuerza de las tortugas
26                 int tempFuerza = fuerza[j];
27                 fuerza[j] = fuerza[j + 1];
28                 fuerza[j + 1] = tempFuerza;
29             }
30         }
31     }
32 }
33
34 int main()
35 {
36     int p, f;
37     int n = 0; // Contador de tortugas
38
39     // Leer datos de entrada de cada tortuga
40     while (cin >> p >> f)
41     {
42         peso[n] = p;
43         fuerza[n] = f;
44         ++n;
45     }
46
47     // Ordenar tortugas por su fuerza para facilitar la construcción de la pila
48     ordenarTortugas(n);
49
50     // Inicializar dp con un valor grande para representar peso imposible
51     for (int i = 0; i <= n; ++i)
52     {
53         dp[i] = PESO_MAXIMO;
54     }
55     dp[0] = 0; // Peso de una pila vacía es 0
56
57     int alturaMaxima = 0; // Máxima altura de la pila de tortugas que se puede construir
58
59     // Intentar agregar cada tortuga a una pila
60     for (int i = 0; i < n; ++i)
61     {
62         // Recorrer dp desde el final para evitar sobrescribir valores previos
63         for (int j = alturaMaxima; j >= 0; --j)
64         {
65             // Verificar si podemos agregar la tortuga actual sin exceder su fuerza
66             if (dp[j] <= fuerza[i] - peso[i])
67             {
68                 // Actualizar el peso mínimo para una pila de altura j+1 si esta combinación es válida
69                 dp[j + 1] = (dp[j + 1] < dp[j] + peso[i]) ? dp[j + 1] : (dp[j] + peso[i]);
70                 // Actualizar la altura máxima alcanzada
71                 alturaMaxima = (alturaMaxima > j + 1) ? alturaMaxima : j + 1;
72             }
73         }
74     }
75
76     // Imprimir la altura máxima de la pila que se puede construir sin que la tortuga ripee
77     cout << alturaMaxima << endl;
78
79     return 0;
80 }
```