SPOJ ELIS

Easy Longest Increasing Subsequence

Prof. Edson Alves – UnB/FGA

Problema

Given a list of numbers A output the length of the longest increasing subsequence. An increasing subsequence is defined as a set $\{i_0, i_1, i_2, i_3, \ldots, i_k\}$ such that $0 \leq i_0 < i_1 < i_2 < i_3 < \ldots < i_k < N$ and $A[i_0] < A[i_1] < A[i_2] < \ldots < A[i_k]$. A longest increasing subsequence is a subsequence with the maximum k (length).

I.e. in the list $\{33, 11, 22, 44\}$ the subsequence $\{33, 44\}$ and $\{11\}$ are increasing subsequences while $\{11, 22, 44\}$ is the longest increasing subsequence.

Entrada e saída

Input

First line contain one number N ($1 \leq N \leq 10$) the length of the list A.

Second line contains N numbers ($1 \leq$ each number ≤ 20), the numbers in the list A separated by spaces.

Output

One line containing the lenght of the longest increasing subsequence in A.

Exemplo de entradas e saídas

Sample Input

```
5
1 4 2 4 3
```

Sample Output

```
3
```

Solução $O(N^2)$

- O título do problema não é um engodo: os limites do problema são tão pequenos que permitem mesmo uma solução de busca completa, avaliando todas as 2^N subsequências de A possíveis
- Uma solução de fácil codificação é a implementação $O(N^2)$ da LIS
- Nela, o estado $lis(i)$ corresponde ao tamanho da maior subsequência crescente que termina no elemento a_i
- O caso baso acontece quando $i = 1$ ($lis(1) = 1$)
- A transição é linear: todos os elementos a_j anteriores ($j < i$) devem ser avaliados
- Assim,

$$lis(i) = \max\{lis(j) + 1, 1\}, \quad \text{para todos } j < i \text{ tais que } a_j < a_i$$

Solução $O(N^2)$

```cpp
#include <bits/stdc++.h>

using namespace std;

int solve(int N, const vector<int>& xs)
{
    vector<int> st(N, 1);

    st[0] = 1;

    for (int i = 1; i < N; ++i)
    {
        for (int j = i - 1; j >= 0; --j)
            if (xs[i] > xs[j])
                st[i] = max(st[i], st[j] + 1);
    }

    return *max_element(st.begin(), st.end());
}
```

Solução $O(N^2)$

```
21 int main()
22 {
23     int N;
24     cin >> N;
25
26     vector<int> xs(N);
27
28     for (int i = 0; i < N; ++i)
29         cin >> xs[i];
30
31     auto ans = solve(N, xs);
32
33     cout << ans << '\n';
34
35     return 0;
36 }
```