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

using namespace std;

template<typename T>
class DST {
public:
  int n;
  function<T(T, T)> f;
  vector<T> v;
  vector<pair<vector<T>, vector<T>>> table[32];

  DST(vector<T> &_v, function<T(T, T)> f) : v(_v), f(f) {
    n = 1;
    while (n < v.size()) {
      n <<= 1;
    }
    v.resize(n);
    int m = n >> 1;
    for (int k = 0; m; k++, m >>= 1) {
      table[k].resize(m);
      int kshift = 1 << k;
      int piv = kshift;
      for (int i = 0; i < m; i++) {
        table[k][i].first.resize(kshift);
        table[k][i].second.resize(kshift);
        table[k][i].first[0] = v[piv - 1];
        table[k][i].second[0] = v[piv];
        for (int j = 1; j < kshift; j++) {
          table[k][i].first[j] = f(table[k][i].first[j - 1], v[piv - 1 - j]);
          table[k][i].second[j] = f(table[k][i].second[j - 1], v[piv + j]);
        }
        piv += 2 << k;
      }
    }
  }

  T query(int l, int r) {
    if (r - l <= 1)
      return v[l];
    r--;
    int k = 31 - __builtin_clz(l ^ r);
    int index = l >> (k + 1);
    int piv = index * (2 << k) + (1 << k);
    return f(table[k][index].first[piv - l - 1],
             table[k][index].second[r - piv]);
  }
};
// ref: http://noshi91.hatenablog.com/entry/2018/05/08/183946
// ref: https://discuss.codechef.com/t/tutorial-disjoint-sparse-table/17404
```

```cpp

int main() {
  int n;
  cin >> n;
  vector<int> v(n);
  for (int i = 0; i < n; i++)
    v[i] = i + 1;
  DST<int> dst(v, [](int a, int b) { return max(a, b); });
  for (int i = 0; i < n; i++) {
    for (int j = i + 1; j <= n; j++) {
      int q = dst.query(i, j);
      assert(q > 0);
    }
  }
  return 0;
}
```