Competitive STL Extensions

Meeting C++ 2018

Fedor Alekseev

Moscow IPT: My pity

Competitive STL Extensions

Good evening everyone! Thanks for having me. My name is Fedor. I'm a student from Moscow.

I'm also doing some competitive programming as a hobby. I have a team called My pity.

Outline

Competitive Programming

Competitive Programming

Engineering is Programming integrated over time?

Standard library

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

Standard library

- `#include <bits/stdc++.h>`
- `std::__gcd` from `<algorithm>`

popcount: number of set bits

```
int main(int argc, const char* argv[]) {
    static_assert(0 == __builtin_popcount(0));  // wow so constexpr
    static_assert(4 == __builtin_popcount(0b1111));
    static_assert(3 == __builtin_popcount(0b100101));
    return __builtin_popcount(argc);
}
```

godbolts to

```
main:
        xor     eax, eax
        popcnt  eax, edi
        ret
```

ctz: Count Trailing Zeros

```
int main(int argc, const char* argv[]) {
    static_assert(32 == __builtin_ctz(0));
    static_assert(0 == __builtin_ctz(0b1111));
    static_assert(2 == __builtin_ctz(0b10100));
    return __builtin_ctz(argc);
}
```

godbolts to

```
main:
        xor     eax, eax
        tzcnt   eax, edi
        ret
```

Also `__builtin_clz(int)` counts leading zeros

SGI STL extensions: power

```
#include <bits/extc++.h>
constexpr int64_t Modulo = 1000000007;
auto multiply_modulo = [](int64_t a, int64_t b) {
  return a * b % Modulo;
};
int64_t identity_element(decltype(multiply_modulo)) {
  return 1;
}
bool fermat_little_theorem_holds(int64_t x) {
  auto inverse = __gnu_cxx::power(x, Modulo - 2, multiply_modulo);
  return 0 == x || 1 == multiply_modulo(x, inverse);
}
```

SGI STL extensions: power

rope

Kool tricks: pbds order statistics tree

pbds gp hash table

kthxbye