NINGBO UNIVERSITY

我可以说中文
如果有副标题的话

第一作者　　第二作者
宁波大学商学院

2013 年 11 月 22 日

宁波大学
NINGBO UNIVERSITY

- ► Use `itemize` a lot.
- ► Use very short sentences or short phrases.

You can create overlays…

- using the `pause` command:
  - First item.

You can create overlays…

- using the `pause` command:
    - First item.
    - Second item.

- using overlay specifications:

- using the general `uncover` command:

You can create overlays…

- using the `pause` command:
  - First item.
  - Second item.
- using overlay specifications:
  - First item.

- using the general `uncover` command:

You can create overlays…

- using the `pause` command:
    - First item.
    - Second item.
- using overlay specifications:
    - First item.
    - Second item.
- using the general `uncover` command:

You can create overlays…

- ► using the `pause` command:
  - ► First item.
  - ► Second item.

- ► using overlay specifications:
  - ► First item.
  - ► Second item.

- ► using the general `uncover` command:
  - ► First item.

宁波大学
NINGBO UNIVERSITY

You can create overlays…

- ▶ using the `pause` command:
  - ▶ First item.
  - ▶ Second item.

- ▶ using overlay specifications:
  - ▶ First item.
  - ▶ Second item.

- ▶ using the general `uncover` command:
  - ▶ First item.
  - ▶ Second item.

动机
　　我们了解到的基本问题
　　前人的工作

Our Results/Contribution
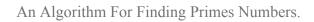　　Main Results
　　Basic Ideas for Proofs/Implementation

```cpp
int main (void)
{
  std::vector<bool> is_prime (100, true);
  for (int i = 2; i < 100; i++)
    if (is_prime[i])
      {
        std::cout << i << " ";
        for (int j = i; j < 100;
             is_prime [j] = false, j+=i);
      }
  return 0;
}
```

```
int main (void)
{
 std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)




 return 0;
}
```
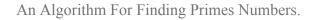
```
int main (void)
{
 std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
    if (is_prime[i])
      {



      }
 return 0;
}
```

```
int main (void)
{
 std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
    if (is_prime[i])
      {
        std::cout << i << " ";
        for (int j = i; j < 100;
             is_prime [j] = false, j+=i);
      }
 return 0;
}
```

```cpp
int main (void)
{
 std::vector<bool> is_prime (100, true);
 for (int i = 2; i < 100; i++)
    if (is_prime[i])
      {
        std::cout << i << " ";
        for (int j = i; j < 100;
             is_prime [j] = false, j+=i);
      }
 return 0;
}
```

Note the use of `std::`.

Example

- ▶ 2 is prime (two divisors: 1 and 2).
- ▶ 3 is prime (two divisors: 1 and 3).
- ▶ 4 is not prime (three divisors: 1, 2, and 4).

宁波大学
NINGBO UNIVERSITY

Theorem

There is no largest prime number and, in addition,

$$\int_\Omega \nabla u \cdot \nabla v = - \int_\Omega u \Delta v + \int_{\partial\Omega} uvn$$

Proof.

1. Suppose p were the largest prime number.

4. Thus $q + 1$ is also prime and greater than p. $\qquad\square$

Theorem

There is no largest prime number and, in addition,

$$\int_\Omega \nabla u \cdot \nabla v = - \int_\Omega u \Delta v + \int_{\partial\Omega} uvn$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.

4. Thus $q + 1$ is also prime and greater than p. $\qquad\square$

Theorem

There is no largest prime number and, in addition,

$$\int_{\Omega} \nabla u \cdot \nabla v = - \int_{\Omega} u \Delta v + \int_{\partial\Omega} uvn$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. Thus $q + 1$ is also prime and greater than p. $\square$

Theorem

There is no largest prime number and, in addition,

$$\int_\Omega \nabla u \cdot \nabla v = -\int_\Omega u \Delta v + \int_{\partial\Omega} uvn$$

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. Thus $q + 1$ is also prime and greater than p. $\square$

The proof used *reductio ad absurdum*.

- The first main message of your talk in one or two lines.
- The second main message of your talk in one or two lines.
- Perhaps a third message, but not more than that.

- Outlook
  - Something you haven't solved.
  - Something else you haven't solved.

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal of This and That, 2(1):50--100, 2000.