This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

0.1 Some theorems and stuff

We now delve right into the proof.

Lemma 0.1 *This is the first lemma of the lecture.*

Proof: The proof is by induction on .... For fun, we throw in a figure.

Figure 0.1: A Fun Figure

This is the end of the proof, which is marked with a little box. ■

0.1.1 A few items of note

Here is an itemized list:

- this is the first item;
- this is the second item.

Here is an enumerated list:

1. this is the first item;
2. this is the second item.

Here is an exercise:

Exercise: Show that P $\neq$ NP.

Here is how to define things in the proper mathematical style. Let f_k be the $AND - OR$ function, defined by

$$f_k(x_1, x_2, \ldots, x_{2^k}) = \begin{cases} x_1 & \text{if } k = 0; \\ AND(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{if } k \text{ is even;} \\ OR(f_{k-1}(x_1, \ldots, x_{2^{k-1}}), f_{k-1}(x_{2^{k-1}+1}, \ldots, x_{2^k})) & \text{otherwise.} \end{cases}$$

Theorem 0.2 *This is the first theorem.*

Proof: This is the proof of the first theorem. We show how to write pseudo-code now.

Consider a comparison between x and y:

```
if x or y or both are in S then
    answer accordingly
else
    Make the element with the larger score (say x) win the comparison
    if F(x) + F(y) < n/(t-1) then
        F(x) ← F(x) + F(y)
        F(y) ← 0
    else
        S ← S ∪ {x}
        r ← r + 1
    endif
endif
```

This concludes the proof. ∎