Lecture 8

16 October 2018

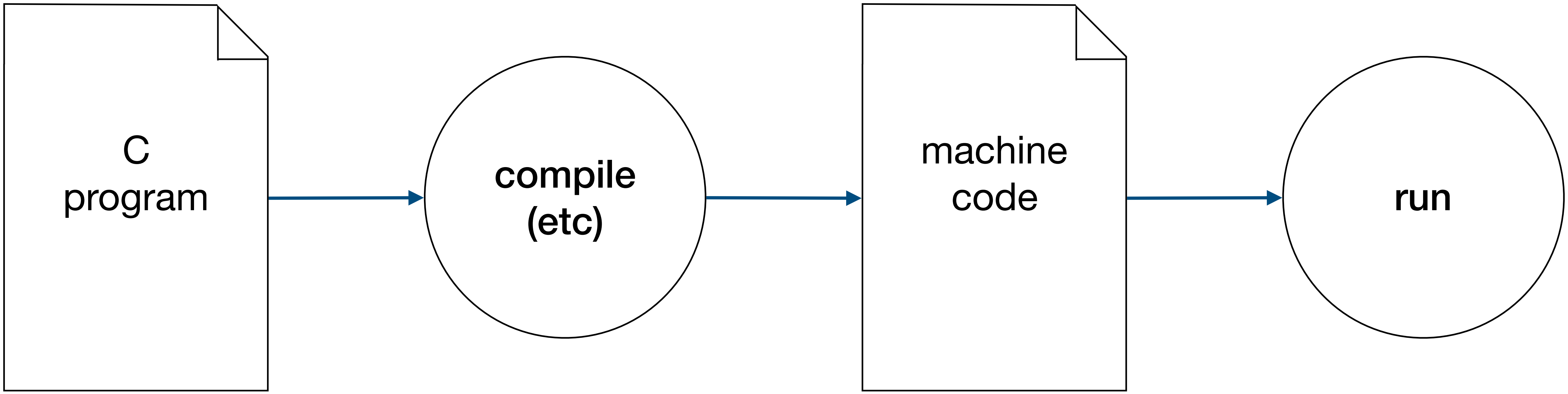
Admin Matters

Unit 20: **C Preprocessor**

Unit 21: **Assert**

Unit 22: **Efficiency**

Previously, on CS1010..

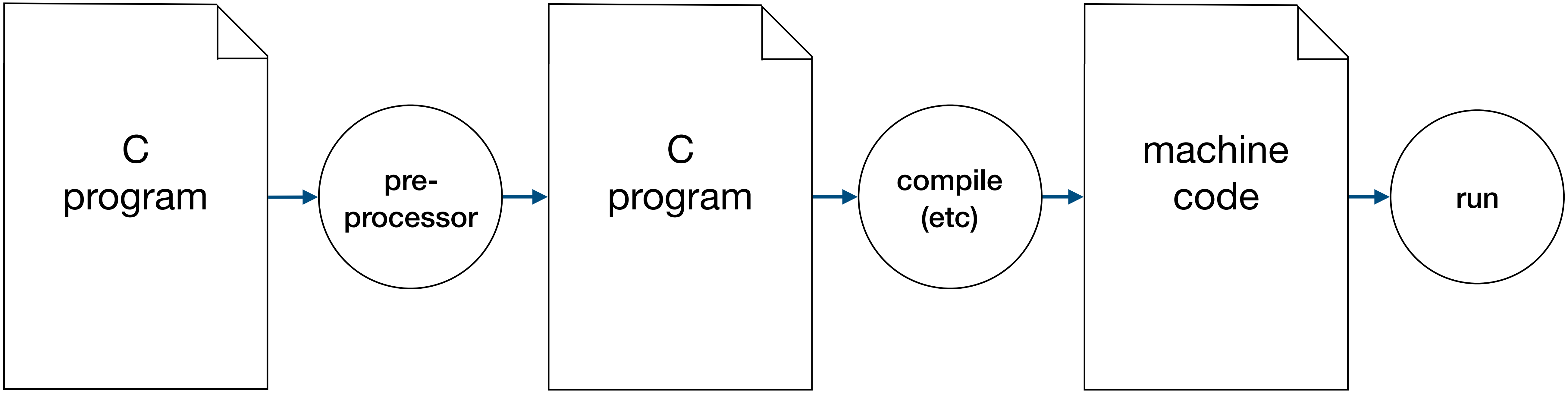


C program

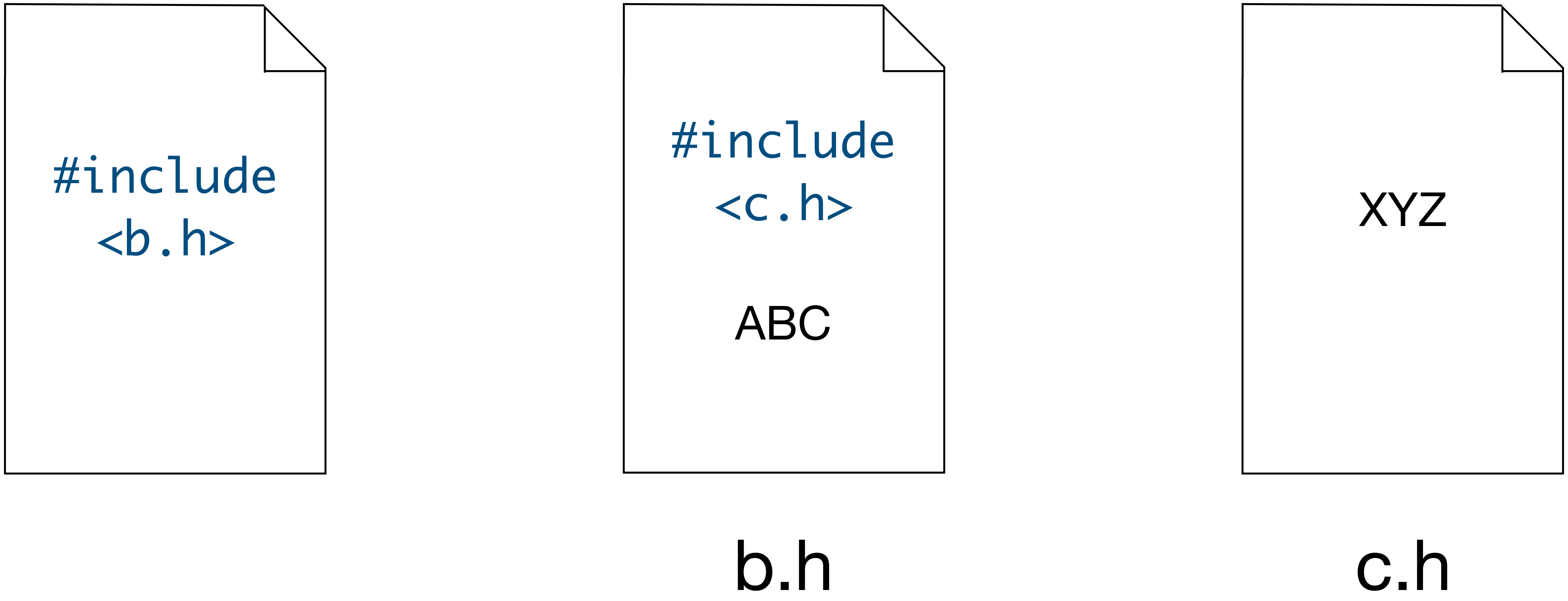
compile (etc)

machine code

run


C
program
pre-
processor
C
program
compile
(etc)
machine
code
run

```
#include <filaneme>
```



```
#include
<b.h>
```

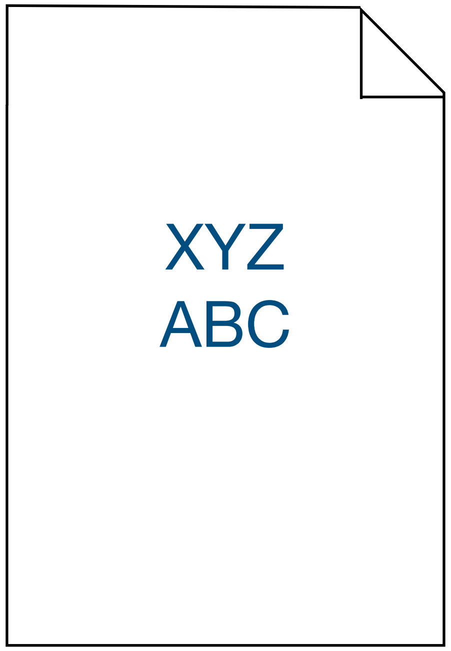
```
#include
<c.h>
```

ABC

b.h

XYZ

c.h



XYZ
ABC

```
#define <id> <token>
```

Macro

```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}
```

```c
void swapd(double *a, double *b) {
  double temp = *a;
  *a = *b;
  *b = temp;
}
```

```c
void swapchar(char *a, char *b) {
  char temp = *a;
  *a = *b;
  *b = temp;
}
```

```c
#include <assert.h>

assert(<condition>);
```

```
long array[4];

  :
// { i >= 0 && i < 4 }
array[i] = 0;
  :
```

```
long array[4];

  :
assert(i >= 0 && i < 4);
array[i] = 0;
  :
```

Efficiency

1. No redundant work

2. No repetition

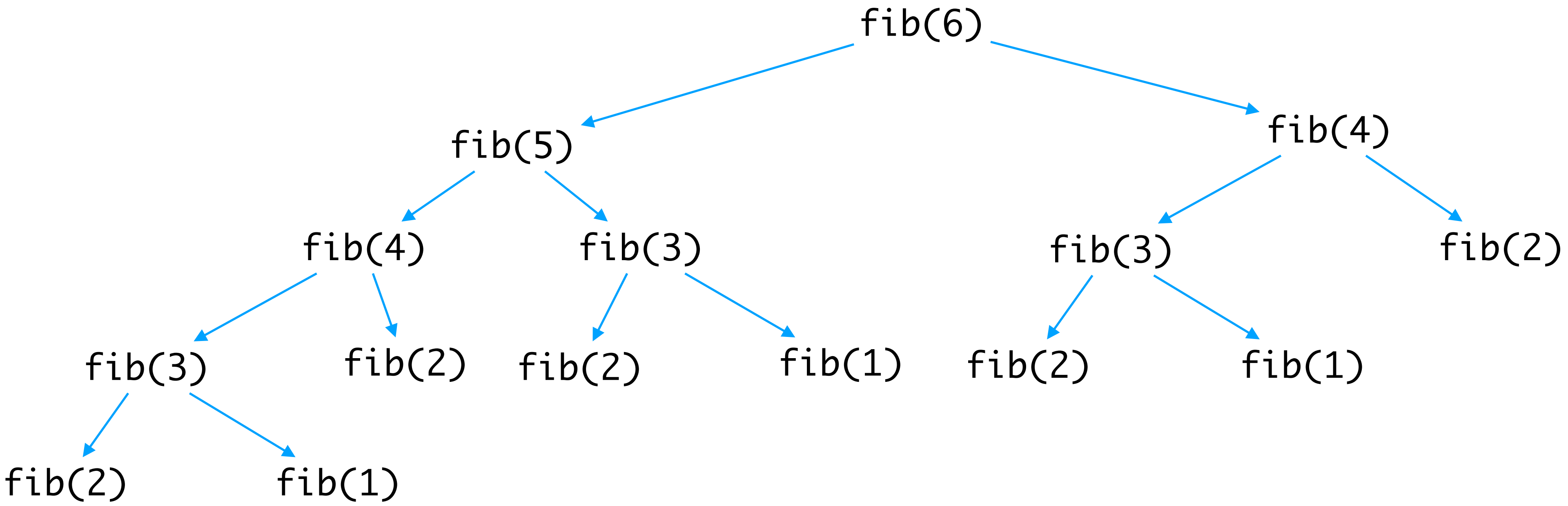
Find the range

- The range of a given list of k integers is the difference between the max and the min values in the list.
- Give an algorithm to find the range

max(L, k) - min(L, k)

```
void find_min_max(long length, long array[length],
    long *min, long *max)
{
  :
}

int main()
{
  long list[10] = {1, 2, 3, 4, -4, 5, 6, -8, 3, 1};
    :
}
```


fib(6)
fib(5)
fib(4)
fib(4)
fib(3)
fib(3)
fib(2)
fib(3)
fib(2)
fib(2)
fib(1)
fib(2)
fib(1)
fib(2)
fib(1)

Big O

rate of growth

f(n) grows faster than g(n),
if we can find
n_o and c such that

$$f(n) > cg(n)$$

for all $n > n_o$

Example: Kendall Tau

In CS1010

1. No redundant work
2. No repetition
3. Smaller Big-O

Assignment 6 onwards

There will be a time limit for your solution