Intro to Time Complexity

Nueva C Compiler | 1 October 2021

Goal:

evaluate how long things take

Approach:
count the number of
operations used

Examples

How many operations?

```
int fn(int n) {
```

```
int x = 1;
x += 1;
```

```
int x = 0;
for (int i=1; i<10; ++i)
{
    x += 1;
}
```

```
return x; }
```

How many operations?

```
int fn(int n) {
```

```
int x = 0;
for (int i=1; i<10; ++i)
{
     x += 1;
}
```

```
return x; }
```

How many operations?

```
int fn(int n) {

                    int x = 0;
                    for (int i=1; i< n; ++i)
                    {
                        x += 1;
                    }

                        return x; }
```

How many operations?

```
int fn(int n) {
```

A polynomial representation:

A + C*n

```
int x = 0;
for (int i=1; i< n; ++i)
{
     x += 1;
}
```

```
return x; }
```

Constant factors are ever changing...

we only care about how the polynomial scales

How many operations?

```
int fn(int n) {
```

```
int x = 1;
x += 1;
```

O(2)

```
int x = 0;
for (int i=1; i<10; ++i)
{
    x += 1;
}
```

O(12)

```
return x; }
```

How many operations?

```
int fn(int n) {
```

```
int x = 1;
x += 1;
```

O(1)

```
int x = 0;
for (int i=1; i<10; ++i)
{
    x += 1;
}
```

O(1)

```
return x; }
```

How many operations?

```
int fn(int n) {
```

A polynomial representation:

A + C*n

```
int x = 0;
for (int i=1; i< n; ++i)
{
    x += 1;
}
```

```
return x; }
```

How many operations?

```
int fn(int n) {
```

A polynomial representation:

$$\lim_{x \to \infty} A + C*n$$

$O(n)$

```
int x = 0;
for (int i=1; i< n; ++i)
{
     x += 1;
}
```

```
return x; }
```

Practice

What's the time complexity?

```
int fn(int n) {
     int x = 1;
     for (int i=1; i<2*n; ++i)
     {

         x += x;

     }
                                    return x; }
```

What's the time complexity?

```
int fn(int n) {
    int x = 1;
    for (int i=1; i<2*n; ++i)
    {
        for (int j=1; j<n; ++j)
        {
            x += x;
        }
    }
                                            return x; }
```

What's the time complexity?

```
int fn(int *arr, int n) {
    for (int i=1; i<n; ++i)
    {
        for (int j=0; j+i<n; ++j)
        {
            if (arr[j] > arr[j+1])
            {
                swap(arr[j], arr[j+1]);
            }
        }
    }                                   return x; }
```

What's the time complexity?

```
int fn(int base, int exp) {
    int ret = 1;
    while (exp)
    {
        if (exp % 2 == 1)
            ret *= base;
        base *= base;
        exp /= 2;
    }

                                        return ret; }
```

What's the time complexity?

```
int fn(int base, int exp) {
    if (exp <= 0) return 1;

    int ret = fn(base, exp/2);
    ret *= ret;
    if (exp % 2 == 1)
        ret *= base;
    return ret;
}
```

What's the time complexity?

```
int fn(int base, int exp, int v=1) {
    if (exp <= 0) return v;

    if (exp % 2 == 1)
        return fn(base*base, exp/2, v*base);
    else
        return fn(base*base, exp/2, v);
}
```

Problems

Given a list of N numbers
in the range 0 to 1e9,
see if x is in the list.

Sort a list of N numbers
in the range 0 to 1e9

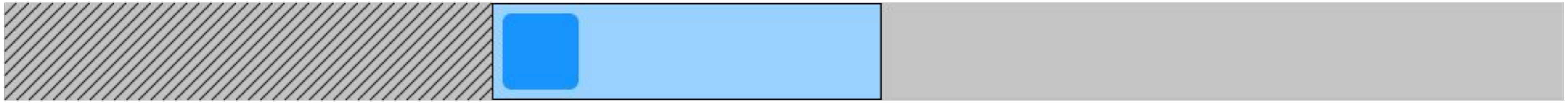
Sort a list of N numbers in the range 0 to 1e4

Given a sorted list of N numbers
in the range 0 to 1e9,
see if x is in the list.

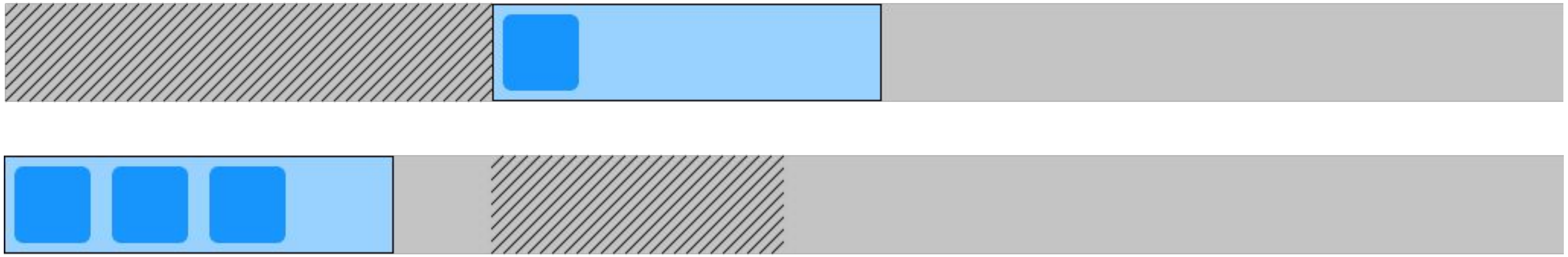
Insert into the front of an array of numbers

Practical Considerations

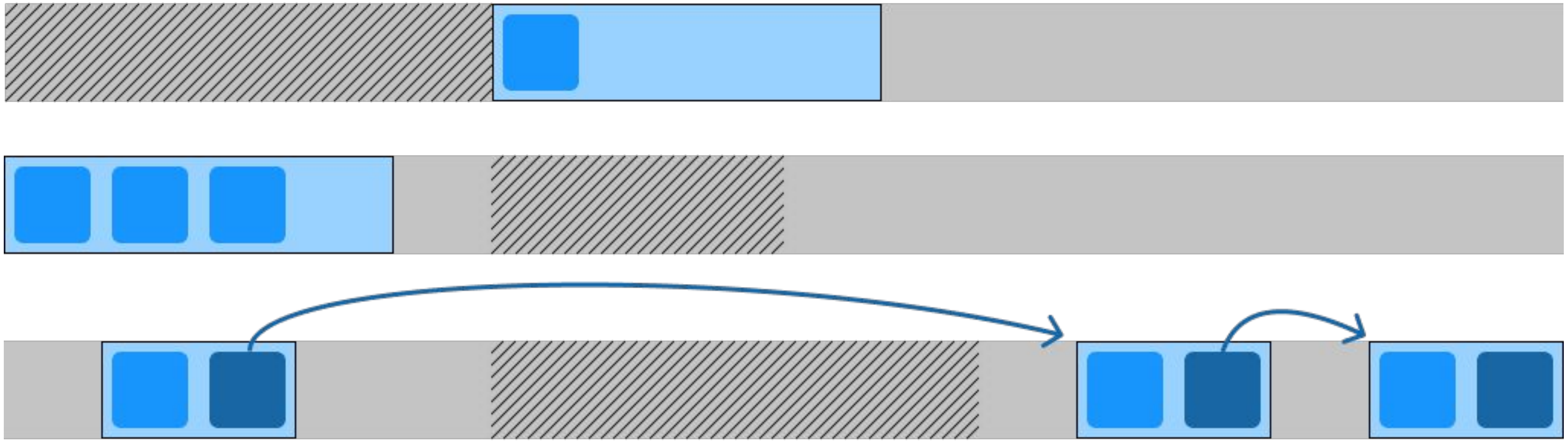
Arrays vs Lists



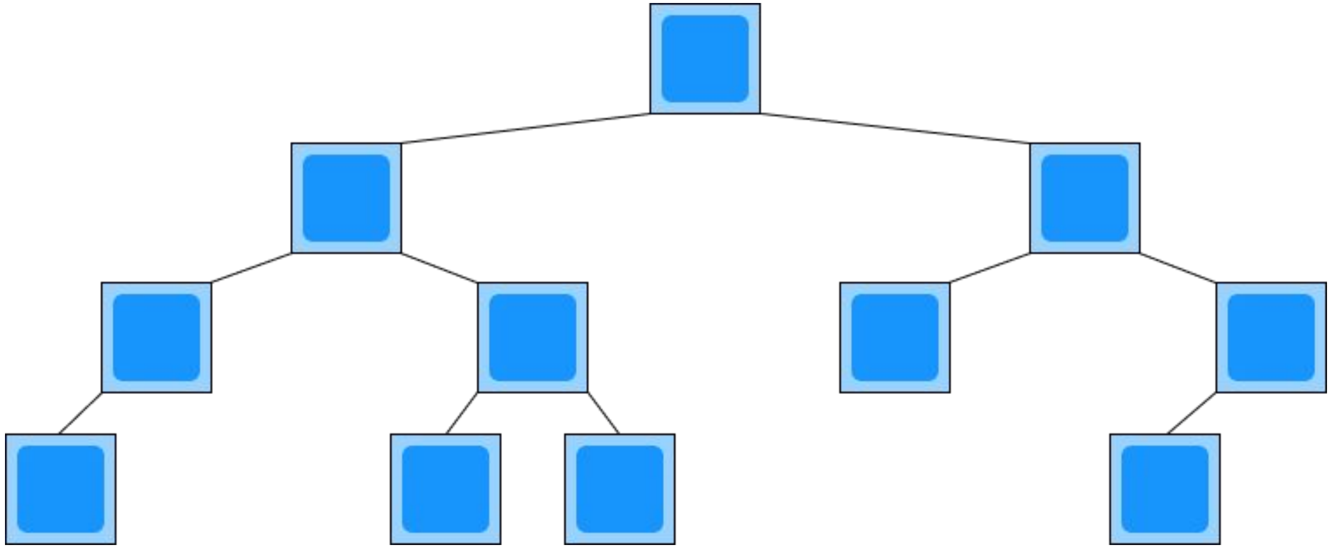
Arrays vs Lists



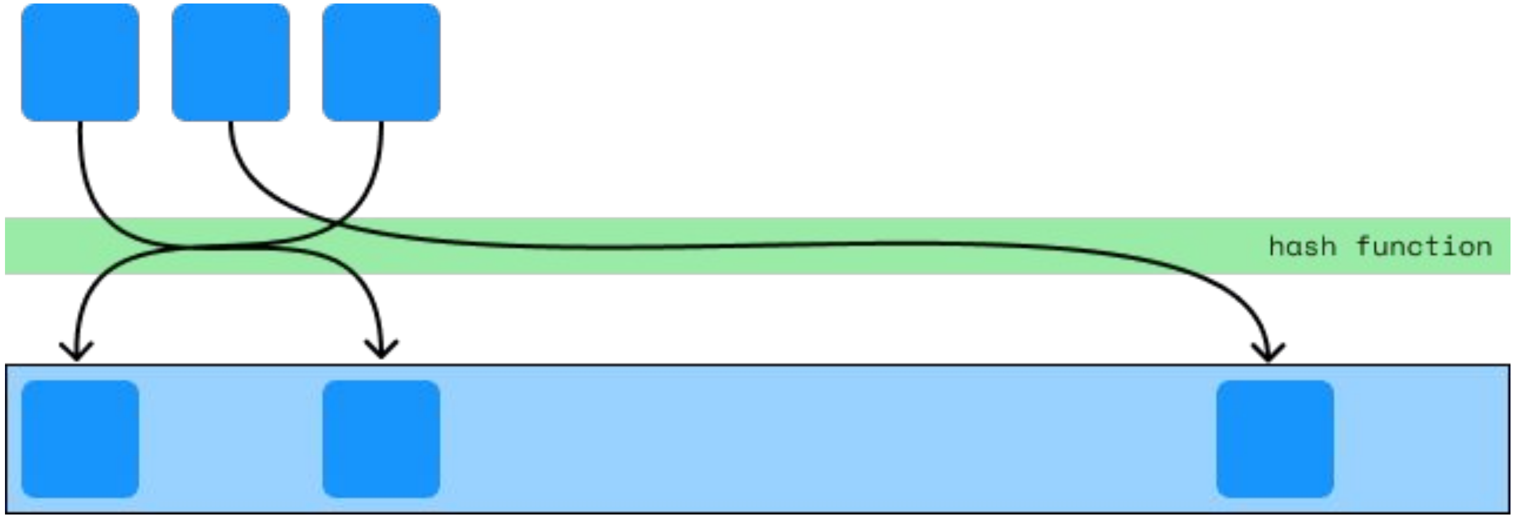
Arrays vs Lists



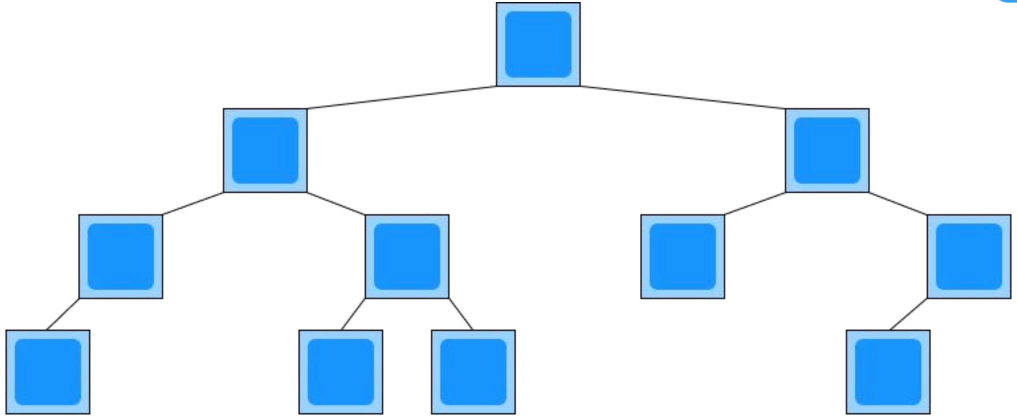
Tree Maps and Hash Maps

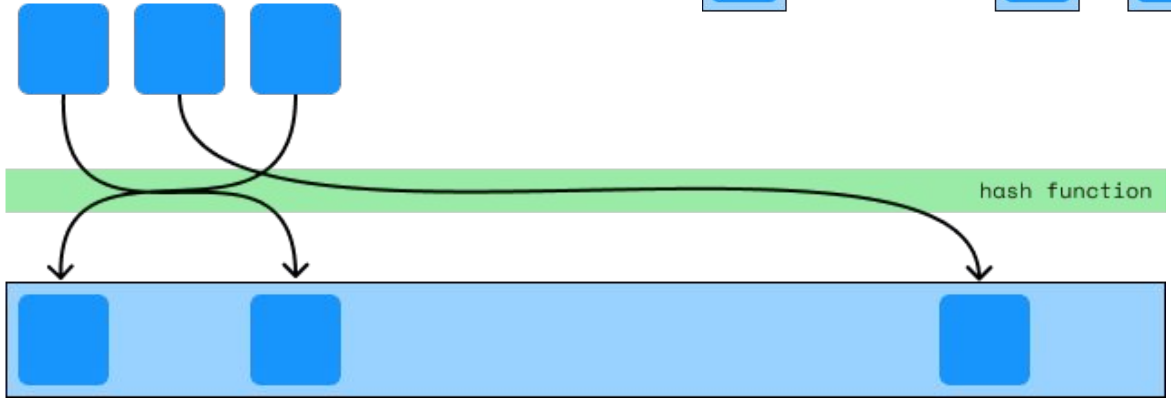


Tree Maps and Hash Maps



Tree Maps and Hash Maps





Constant Factors **Recursive Frames**

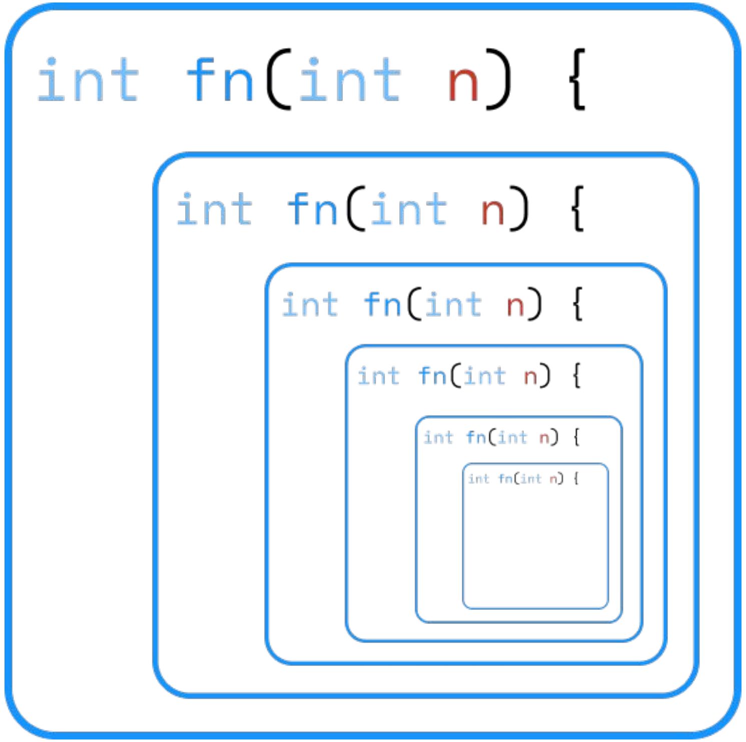
```
int fn(int n) {
```

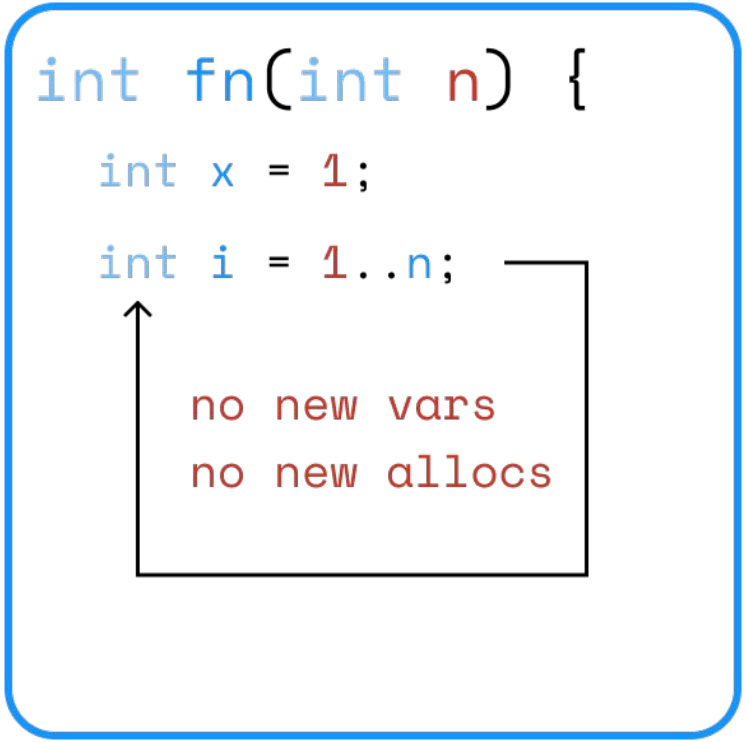
```
if (n == 0) return 1;
int x = fn(n-1) * 2;
```

```
int x = 1;
for (int i=0; i<n; ++i)
{
    x *= 2;
}
```

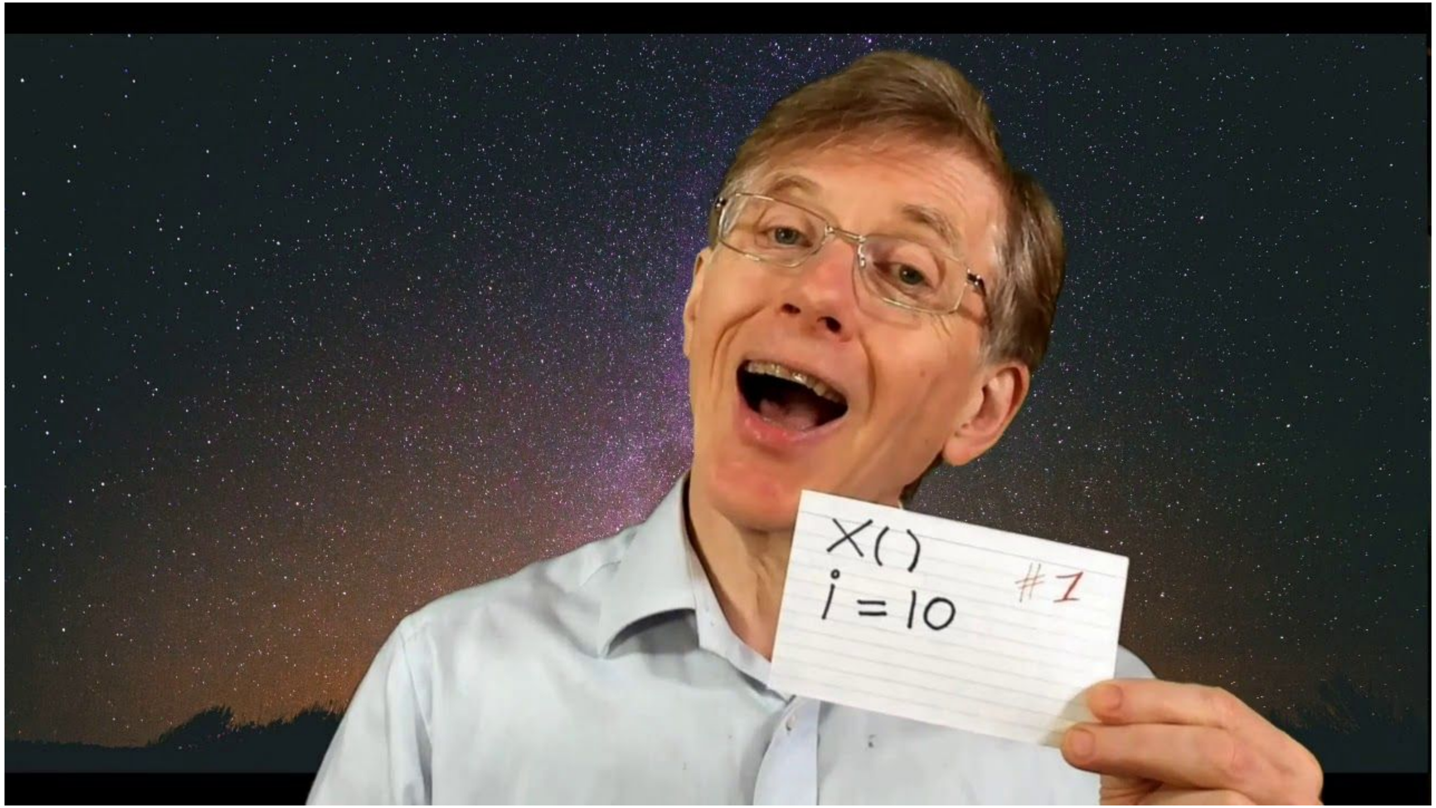
```
return x; }
```

Constant Factors **Recursive Frames**

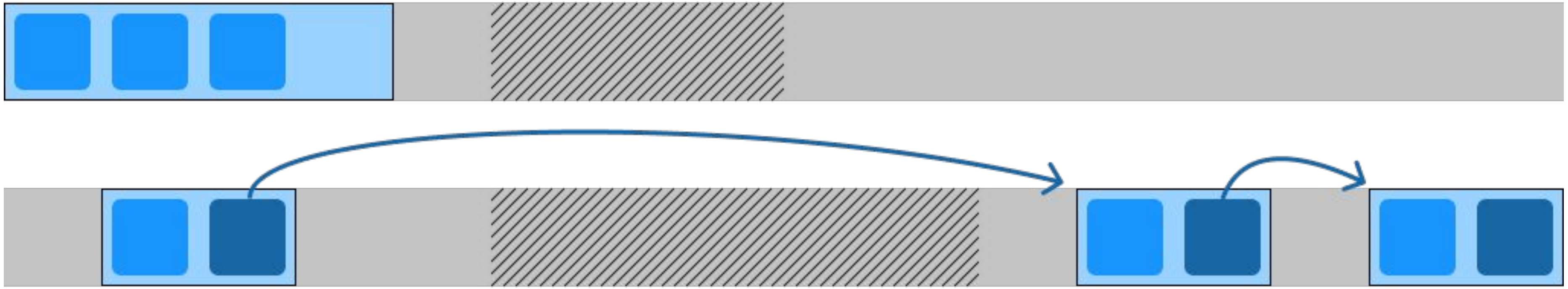




Constant Factors **Recursive Frames**



Constant Factors **Memory Locality**



Constant Factors **Branch Conditions**

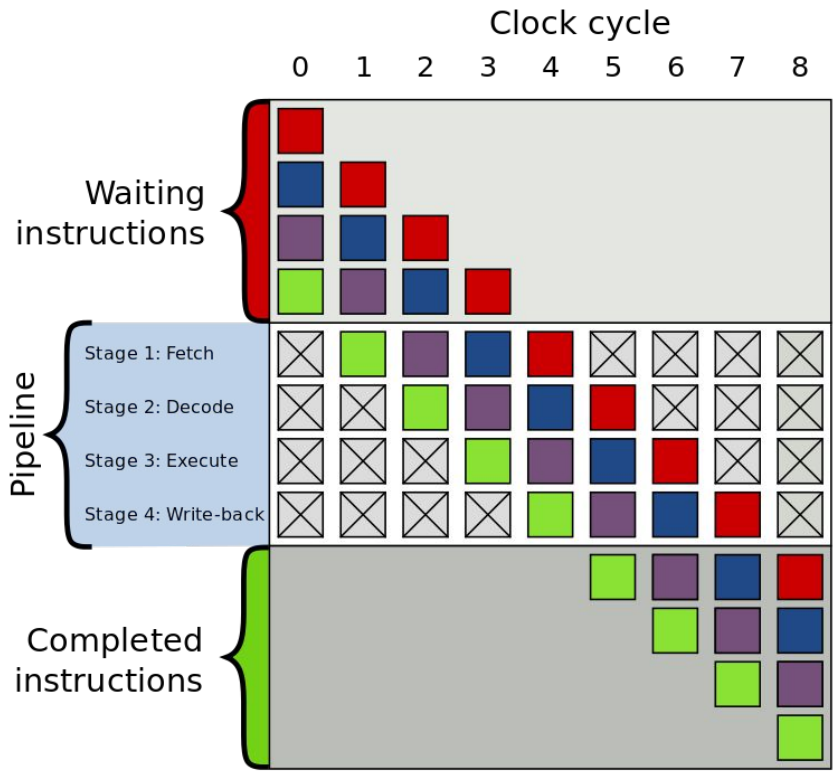
```
int fn(bool x) {
```

```
if (x)
    for (int i=1; i<10; ++i)
    {
        doSomething();
    }
```

```
for (int i=1; i<10; ++i)
{
    if (x)
        doSomething();
}
```

```
}
```

Constant Factors **Branch Conditions**



PROFILE YOUR CODE