1 `main` — MIR Walkthrough

Purpose: TODO: Describe why this walkthrough exists

1.1 Source Context

```
    let b:u32 = 4294967294 + 1;
    assert!(a == b)
}
```

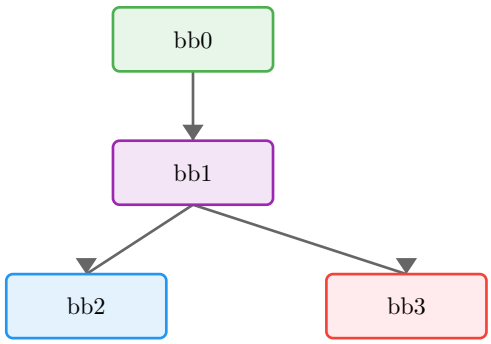
1.2 Function Overview

- **Function:** `main`
- **Basic blocks:** 4
- **Return type:** `()`
- **Notable properties:**
  - Contains panic path
  - Uses checked arithmetic
  - Contains assertions
  - Has conditional branches

1.3 Locals

Local	Type	Notes
0	`()`	Return place
1	`u32`	
2	`(u32, bool)`	
3	`bool`	
4	`u32`	
5	`!`	

1.4 Control-Flow Overview



1.5 Basic Blocks

1.5.1 bb0 — entry

Entry point of the function.

MIR	Annotation
`\_2 = checked(-2 + 1)`	Checked Add (may panic)
`→ assert(move \_2.1 == false) → bb1`	Panic if move _2.1 is true

1.5.2 bb1 — branch point

MIR	Annotation
`\_1 = move \_2.0`	Move value
`\_4 = -1`	Load constant
`\_3 = move \_4 == \_1`	Equal operation
`→ switch(move \_3) \[0→bb3; else→bb2\]`	Branch on move _3

1.5.3 bb2 — return / success

Normal return path.

MIR	Annotation
`→ return`	Return from function

1.5.4 bb3 — panic path

Panic/diverging path.

MIR	Annotation
`→ \_5 = panic(\[16 bytes\])`	Call panic

1.6 Key Observations

TODO: Add bullet points summarizing what this MIR teaches

-
-

1.7 Takeaways

TODO: One or two sentences to generalize this example