CSE 503

Software Engineering

Winter 2021

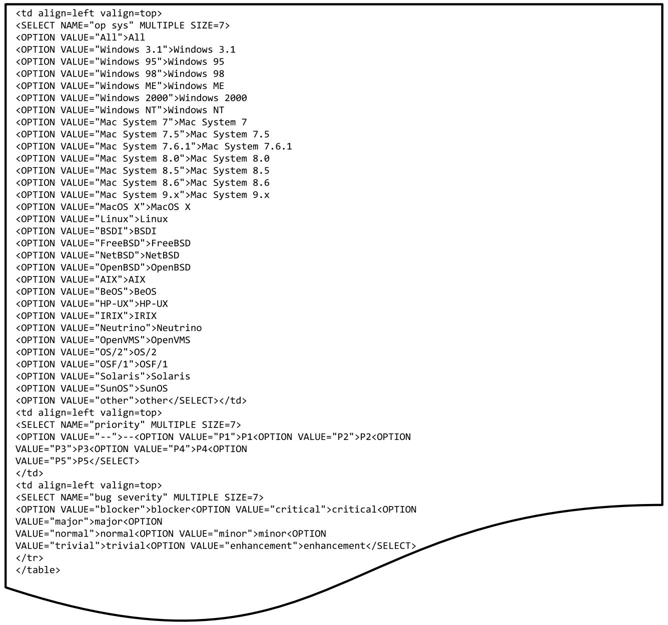
Delta Debugging

February 03, 2021

Today

- Delta Debugging
  - Motivating examples
  - Live demo
  - A little quiz
  - Discussion

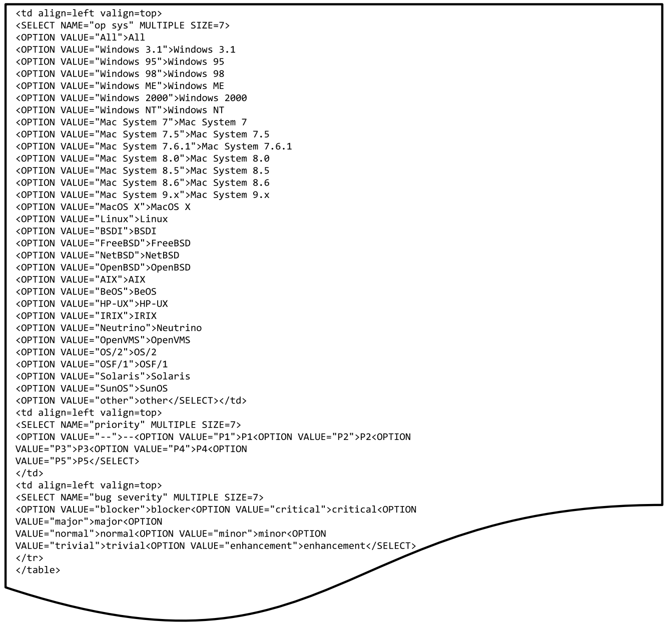
This is a crashing test case



```
<td align=left valign=top>
<SELECT NAME="op sys" MULTIPLE SIZE=7>
<OPTION VALUE="All">All
<OPTION VALUE="Windows 3.1">Windows 3.1
<OPTION VALUE="Windows 95">Windows 95
<OPTION VALUE="Windows 98">Windows 98
<OPTION VALUE="Windows ME">Windows ME
<OPTION VALUE="Windows 2000">Windows 2000
<OPTION VALUE="Windows NT">Windows NT
<OPTION VALUE="Mac System 7">Mac System 7
<OPTION VALUE="Mac System 7.5">Mac System 7.5
<OPTION VALUE="Mac System 7.6.1">Mac System 7.6.1
<OPTION VALUE="Mac System 8.0">Mac System 8.0
<OPTION VALUE="Mac System 8.5">Mac System 8.5
<OPTION VALUE="Mac System 8.6">Mac System 8.6
<OPTION VALUE="Mac System 9.x">Mac System 9.x
<OPTION VALUE="MacOS X">MacOS X
<OPTION VALUE="Linux">Linux
<OPTION VALUE="BSDI">BSDI
<OPTION VALUE="FreeBSD">FreeBSD
<OPTION VALUE="NetBSD">NetBSD
<OPTION VALUE="OpenBSD">OpenBSD
<OPTION VALUE="AIX">AIX
<OPTION VALUE="BeOS">BeOS
<OPTION VALUE="HP-UX">HP-UX
<OPTION VALUE="IRIX">IRIX
<OPTION VALUE="Neutrino">Neutrino
<OPTION VALUE="OpenVMS">OpenVMS
<OPTION VALUE="OS/2">OS/2
<OPTION VALUE="OSF/1">OSF/1
<OPTION VALUE="Solaris">Solaris
<OPTION VALUE="SunOS">SunOS
<OPTION VALUE="other">other</SELECT></td>
<td align=left valign=top>
<SELECT NAME="priority" MULTIPLE SIZE=7>
<OPTION VALUE="--">--<OPTION VALUE="P1">P1<OPTION VALUE="P2">P2<OPTION
VALUE="P3">P3<OPTION VALUE="P4">P4<OPTION
VALUE="P5">P5</SELECT>
</td>
<td align=left valign=top>
<SELECT NAME="bug severity" MULTIPLE SIZE=7>
<OPTION VALUE="blocker">blocker<OPTION VALUE="critical">critical<OPTION
VALUE="major">major<OPTION
VALUE="normal">normal<OPTION VALUE="minor">minor<OPTION
VALUE="trivial">trivial<OPTION VALUE="enhancement">enhancement</SELECT>
</tr>
</table>
```

- Crashed Mozilla
- How would you debug the problem?

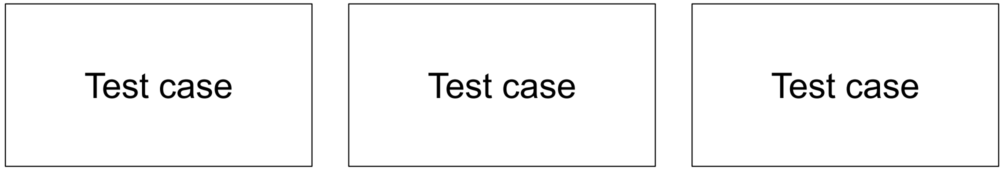
This is a crashing test case



```
<td align=left valign=top>
<SELECT NAME="op sys" MULTIPLE SIZE=7>
<OPTION VALUE="All">All
<OPTION VALUE="Windows 3.1">Windows 3.1
<OPTION VALUE="Windows 95">Windows 95
<OPTION VALUE="Windows 98">Windows 98
<OPTION VALUE="Windows ME">Windows ME
<OPTION VALUE="Windows 2000">Windows 2000
<OPTION VALUE="Windows NT">Windows NT
<OPTION VALUE="Mac System 7">Mac System 7
<OPTION VALUE="Mac System 7.5">Mac System 7.5
<OPTION VALUE="Mac System 7.6.1">Mac System 7.6.1
<OPTION VALUE="Mac System 8.0">Mac System 8.0
<OPTION VALUE="Mac System 8.5">Mac System 8.5
<OPTION VALUE="Mac System 8.6">Mac System 8.6
<OPTION VALUE="Mac System 9.x">Mac System 9.x
<OPTION VALUE="MacOS X">MacOS X
<OPTION VALUE="Linux">Linux
<OPTION VALUE="BSDI">BSDI
<OPTION VALUE="FreeBSD">FreeBSD
<OPTION VALUE="NetBSD">NetBSD
<OPTION VALUE="OpenBSD">OpenBSD
<OPTION VALUE="AIX">AIX
<OPTION VALUE="BeOS">BeOS
<OPTION VALUE="HP-UX">HP-UX
<OPTION VALUE="IRIX">IRIX
<OPTION VALUE="Neutrino">Neutrino
<OPTION VALUE="OpenVMS">OpenVMS
<OPTION VALUE="OS/2">OS/2
<OPTION VALUE="OSF/1">OSF/1
<OPTION VALUE="Solaris">Solaris
<OPTION VALUE="SunOS">SunOS
<OPTION VALUE="other">other</SELECT></td>
<td align=left valign=top>
<SELECT NAME="priority" MULTIPLE SIZE=7>
<OPTION VALUE="--">--<OPTION VALUE="P1">P1<OPTION VALUE="P2">P2<OPTION
VALUE="P3">P3<OPTION VALUE="P4">P4<OPTION
VALUE="P5">P5</SELECT>
</td>
<td align=left valign=top>
<SELECT NAME="bug severity" MULTIPLE SIZE=7>
<OPTION VALUE="blocker">blocker<OPTION VALUE="critical">critical<OPTION
VALUE="major">major<OPTION
VALUE="normal">normal<OPTION VALUE="minor">minor<OPTION
VALUE="trivial">trivial<OPTION VALUE="enhancement">enhancement</SELECT>
</tr>
</table>
```

- Crashed Mozilla
- How would you debug the problem?
- A minimal test case is: <SELECT>
- Can we automate the process of minimizing test cases?
- What’s the naive approach for an optimal solution?

Minimizing test cases

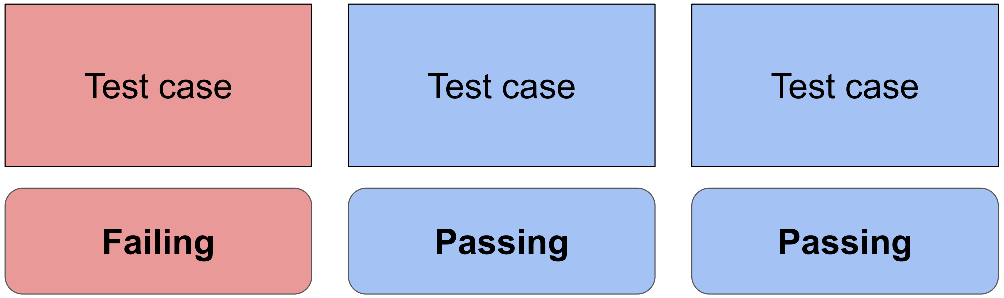


Test case

Test case

Test case

Minimizing test cases



Test case

Test case

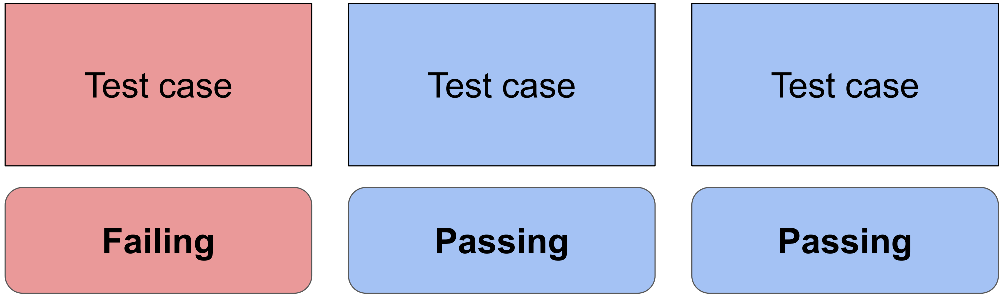
Test case

Failing

Passing

Passing

Minimizing test cases



Test case

Test case

Test case

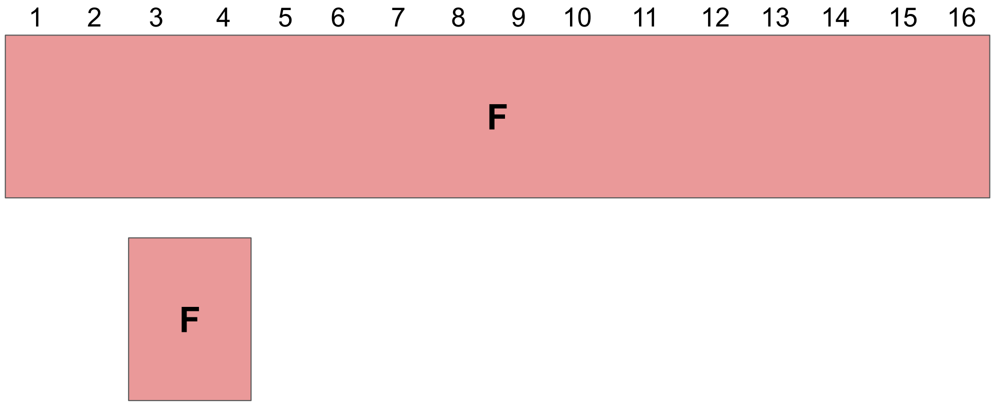
Failing

Passing

Passing

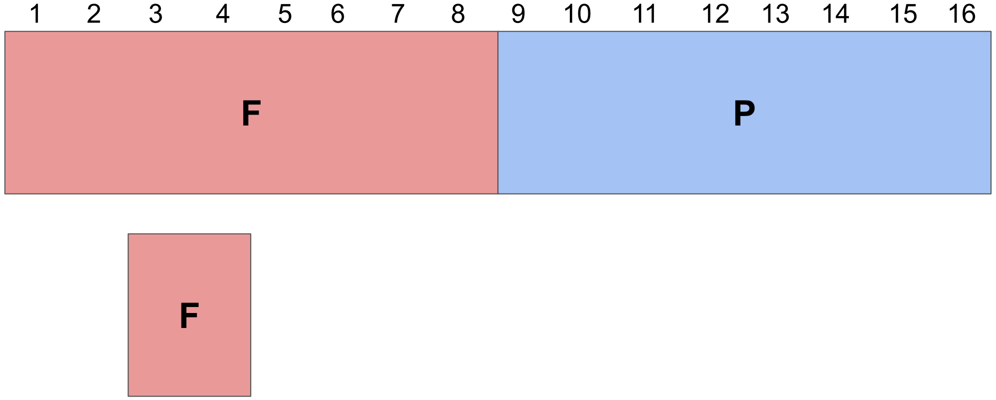
Goal: Minimize the failing test case

The happy path: binary search



Failing test with 16 lines.
The minimal test has 2 lines.

The happy path: binary search

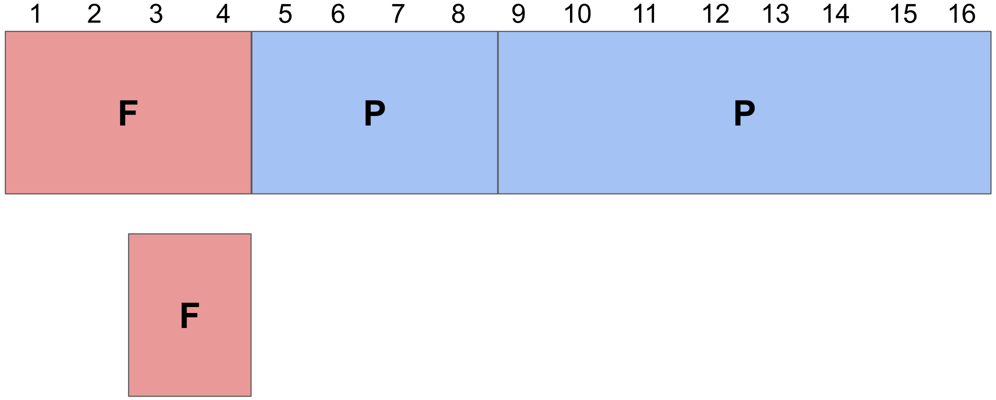


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

F P

F

The happy path: binary search

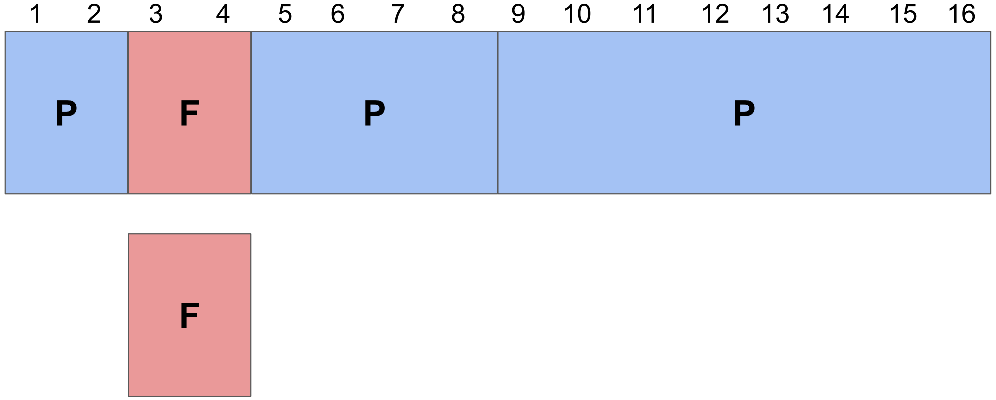


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

F P P

F

The happy path: binary search

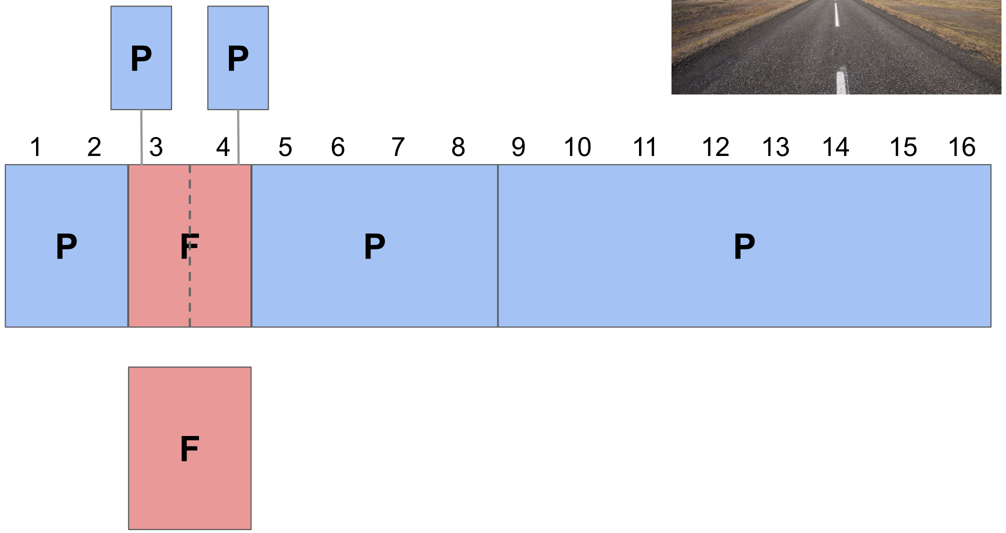


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

P F P P

F

The happy path: binary search



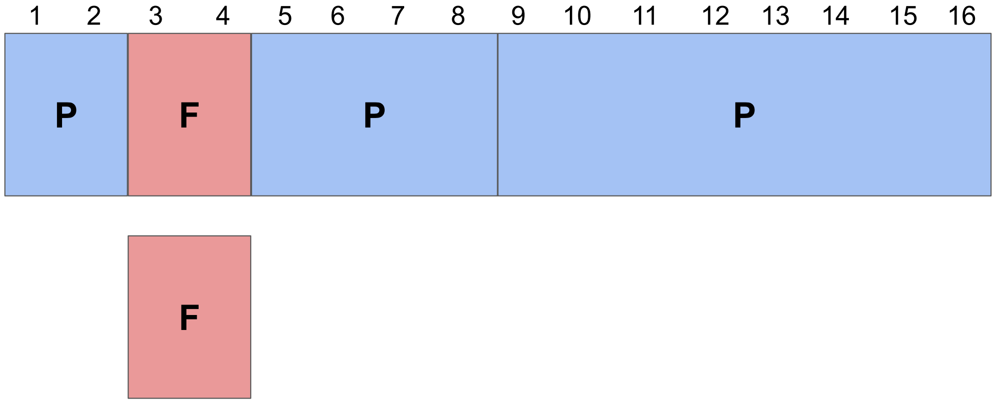
P P

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

P F P P

F

The happy path: binary search



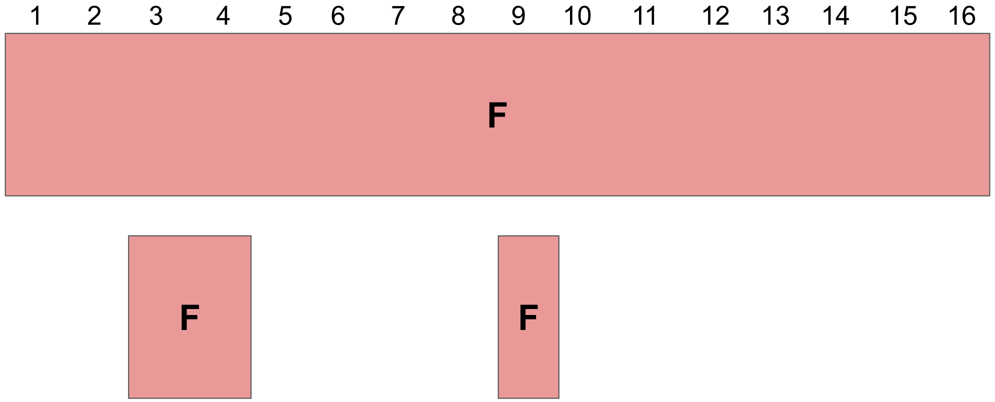
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

P F P P

F

Successfully minimized the failing test to 2 lines

The not so happy path...



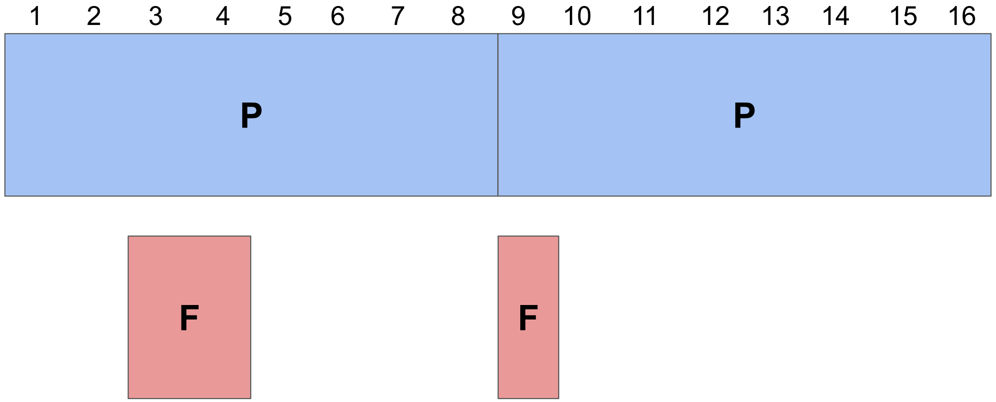
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

F

F F

Suppose the failure pattern is more complex.

The not so happy path...



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

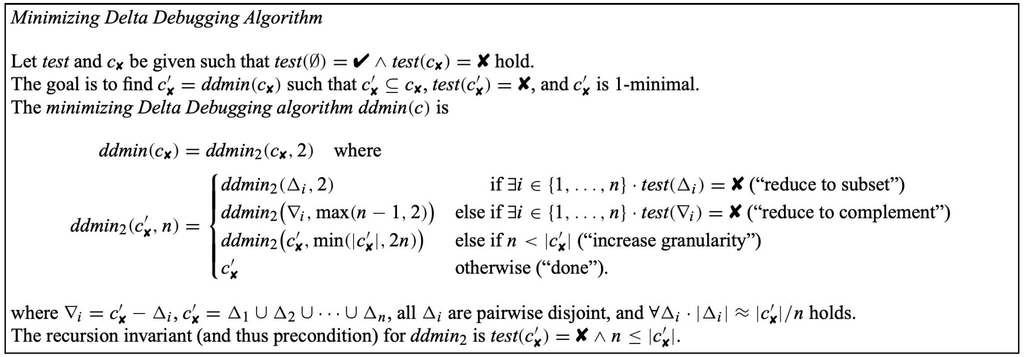
P P

F F

Binary search does not give optimal results.

Delta debugging: binary search + X

The DD algorithm



Minimizing Delta Debugging Algorithm

Let *test* and $c_{\text{✘}}$ be given such that $test(\emptyset) = \text{✔} \wedge test(c_{\text{✘}}) = \text{✘}$ hold.
The goal is to find $c'_{\text{✘}} = ddmin(c_{\text{✘}})$ such that $c'_{\text{✘}} \subseteq c_{\text{✘}}$, $test(c'_{\text{✘}}) = \text{✘}$, and $c'_{\text{✘}}$ is 1-minimal.
The *minimizing Delta Debugging algorithm* $ddmin(c)$ is

$$ddmin(c_{\text{✘}}) = ddmin_2(c_{\text{✘}}, 2) \quad \text{where}$$

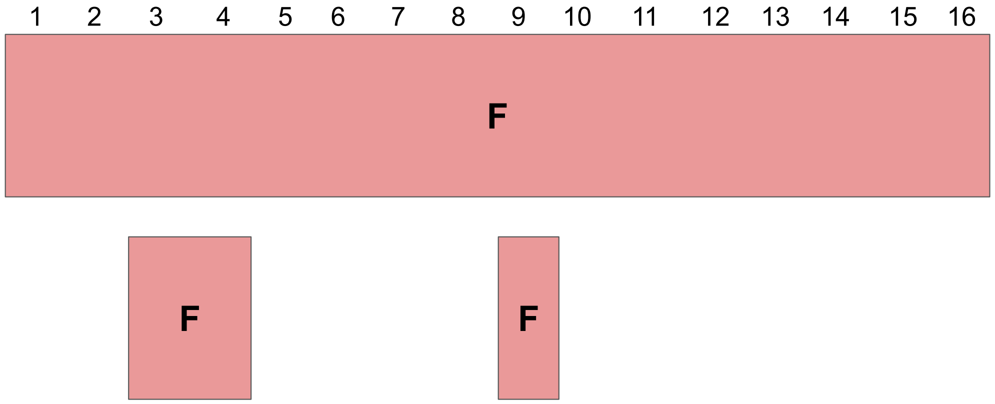
$$ddmin_2(c'_{\text{✘}}, n) = \begin{cases} ddmin_2(\Delta_i, 2) & \text{if } \exists i \in \{1, \ldots, n\} \cdot test(\Delta_i) = \text{✘} \text{ (“reduce to subset”)} \\ ddmin_2\big(\nabla_i, \max(n-1, 2)\big) & \text{else if } \exists i \in \{1, \ldots, n\} \cdot test(\nabla_i) = \text{✘} \text{ (“reduce to complement”)} \\ ddmin_2\big(c'_{\text{✘}}, \min(|c'_{\text{✘}}|, 2n)\big) & \text{else if } n < |c'_{\text{✘}}| \text{ (“increase granularity”)} \\ c'_{\text{✘}} & \text{otherwise (“done”).} \end{cases}$$

where $\nabla_i = c'_{\text{✘}} - \Delta_i$, $c'_{\text{✘}} = \Delta_1 \cup \Delta_2 \cup \cdots \cup \Delta_n$, all Δ_i are pairwise disjoint, and $\forall \Delta_i \cdot |\Delta_i| \approx |c'_{\text{✘}}|/n$ holds.
The recursion invariant (and thus precondition) for $ddmin_2$ is $test(c'_{\text{✘}}) = \text{✘} \wedge n \leq |c'_{\text{✘}}|$.

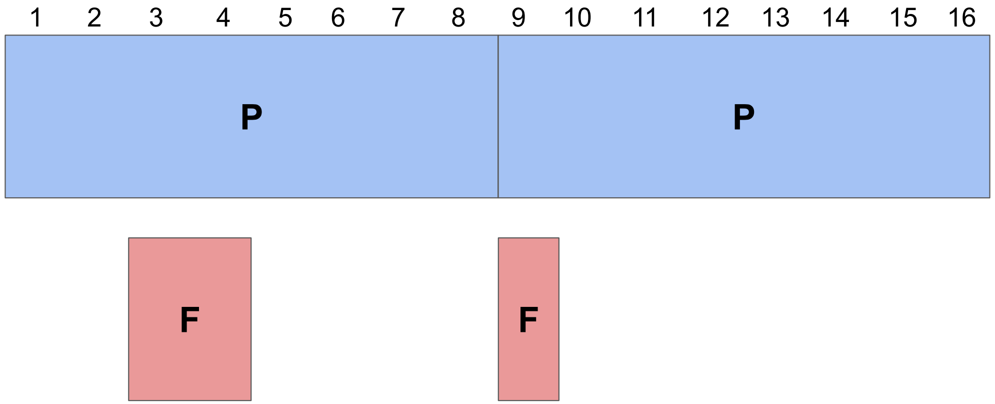
Four basic steps:

1. Test each subset
2. Test each complement
3. Increase granularity
4. Reduce

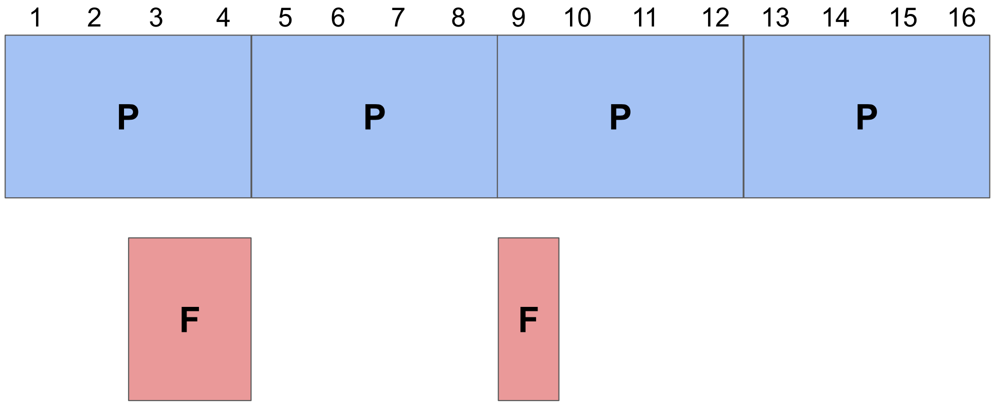
Delta Debugging: mostly binary search



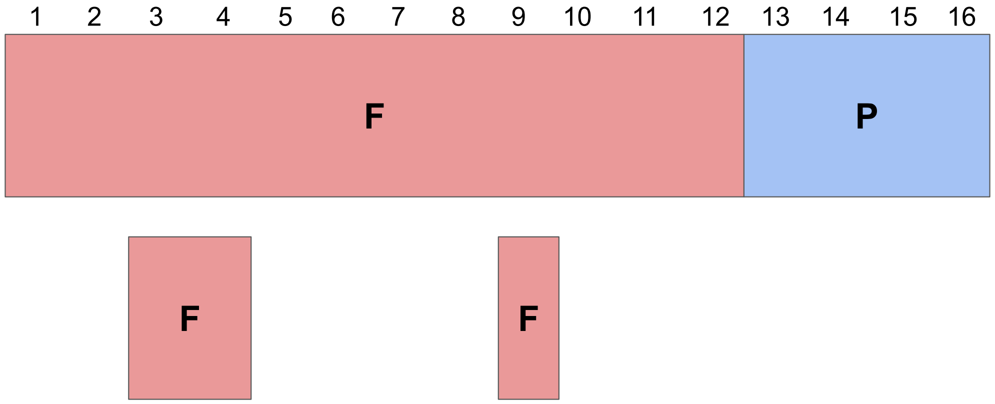
Delta Debugging: mostly binary search



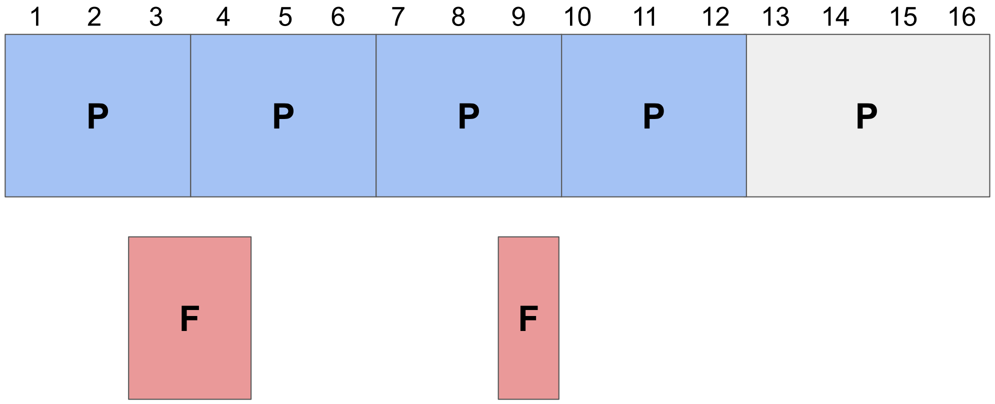
Delta Debugging: granularity



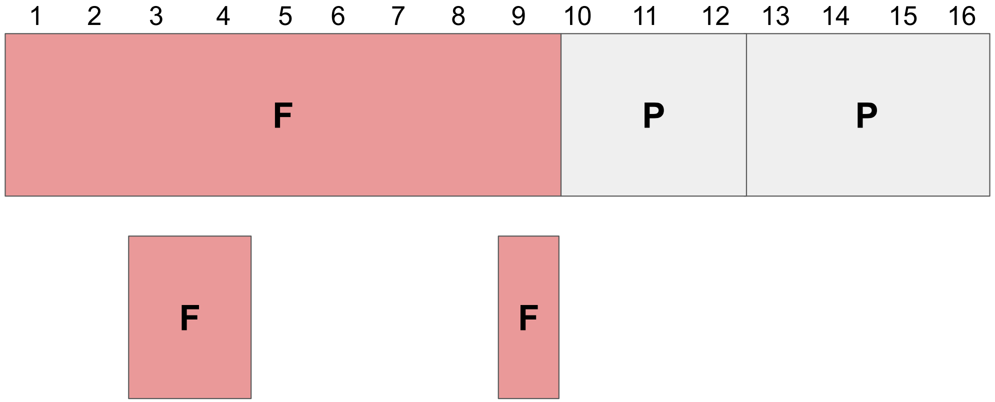
Delta Debugging: complements



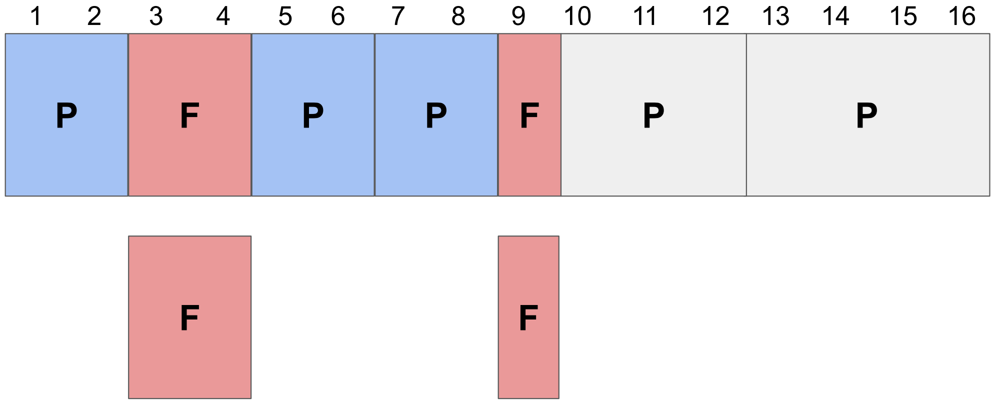
Delta Debugging: reduce



Delta Debugging: reduce



Delta Debugging: 1-minimality



Failure inputs must be deterministic and monotone.

Delta debugging: live examples

A little quiz

Program and initial test case

- Program *P* takes as **input a String of a$_s$ and b$_s$.**
- ***P* crashes** whenever the input String contains an **even number of a$_s$ AND** an **odd number of b$_s$.**
- Assume **character-level** granularity.
- **Initial** crashing **test case** is: **babab.**

Determine the following test cases (using DD)

1. Smallest
2. Local minimum but not smallest
3. 1-minimal of size 3

A little quiz

Program and initial test case

- Program *P* takes as **input a String of a$_s$ and b$_s$.**
- ***P* crashes** whenever the input String contains an **even number of a$_s$ AND** an **odd number of b$_s$.**
- Assume **character-level** granularity.
- **Initial** crashing **test case** is: **babab.**

Determine the following test cases (using DD)

1. Smallest **b**
2. Local minimum but not smallest **NONE**
3. 1-minimal of size 3 **aab**

A little quiz

Program and initial test case

- Program *P* takes as **input an Array of integers** a.
- ***P* crashes** whenever **a contains 42**.
- **Initial** crashing **test case** is: **2424**

Complete the following table

Iteration	n	input	$\triangle_1, \ldots, \triangle_n$ $\nabla_1, \ldots, \nabla_n$
1		2424	

A little quiz

Program and initial test case

- Program *P* takes as **input an Array of integers** a.
- ***P* crashes** whenever a **contains 42**.
- **Initial** crashing **test case** is: **2424**

Complete the following table

Iteration	n	input	$\triangle_1, \ldots, \triangle_n$ $\nabla_1, \ldots, \nabla_n$
1	2	2424	24, (24)
2	4	2424	2, 4, (2), (4), **424**, 224, 244, 242
3	3	424	(4), (2), (4), (24), 44, **42**
4	2	42	(4),(2)

Delta debugging: summary

Discussion

- Non-deterministic programs
- Input structure and granularity
- Monotonicity
- Complexity