

# Fly Space-A

## Implementation Details



[github.com/flyspacea](https://github.com/flyspacea)

06JAN18  
Anson Liu  
[ansonliu.com](http://ansonliu.com)

# What is Fly Space-A

Fly Space-A is computer program that converts visual Space-A flight schedule slides to text data so the military travelers can easily find the flights they need versus visiting every Facebook Space-A terminal page.

Space Available flight is a privilege for members of the US DoD to be transported on DoD aircraft when capacity allows.

Fly Space-A consists of a backend server that processes the photo slides and frontend client web/iOS/Android app for normal users.

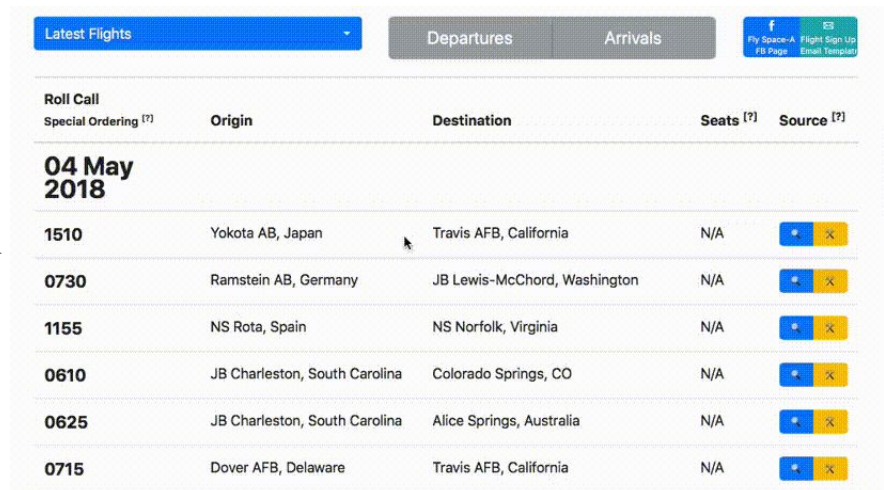


**DEPARTURES**  
**JOINT BASE PEARL HARBOR-HICKAM**  
**SUNDAY, 11 MAR 2018**

ROLL CALL	DESTINATION	SEATS
0510	YOKOTA AB, JAPAN	SP
0610	ELMENDORF AFB, AK WESTOVER ARB, MA	SP
0610	EIELSON AFB, AK FAIRCHILD AFB, WA	SP
0740	KADENA AB, JAPAN	SP
0820	KWAJALEIN ATOLL	41T
1041	TRAVIS AFB, CA	SP

Destinations listed in alphabetical order

```
[[{"Roll Call": "0510", "Origin": "Yokota AB, Japan", "Destination": "Travis AFB, California", "Seats": "SP"}, {"Roll Call": "0610", "Origin": "Elmendorf AFB, AK", "Destination": "JB Lewis-McChord, Washington", "Seats": "SP"}, {"Roll Call": "0610", "Origin": "Westover AFB, MA", "Destination": "NS Norfolk, Virginia", "Seats": "SP"}, {"Roll Call": "0610", "Origin": "Eielson AFB, AK", "Destination": "Colorado Springs, CO", "Seats": "SP"}, {"Roll Call": "0610", "Origin": "Fairchild AFB, WA", "Destination": "Alice Springs, Australia", "Seats": "SP"}, {"Roll Call": "0740", "Origin": "Kadena AB, Japan", "Destination": "Travis AFB, California", "Seats": "SP"}, {"Roll Call": "0820", "Origin": "Kwajalein Atoll", "Destination": "Travis AFB, California", "Seats": "41T"}, {"Roll Call": "1041", "Origin": "Travis AFB, CA", "Destination": "Travis AFB, California", "Seats": "SP"}]]
```



Latest Flights

Departures Arrivals

Roll Call

Special Ordering [?]

Origin

Destination

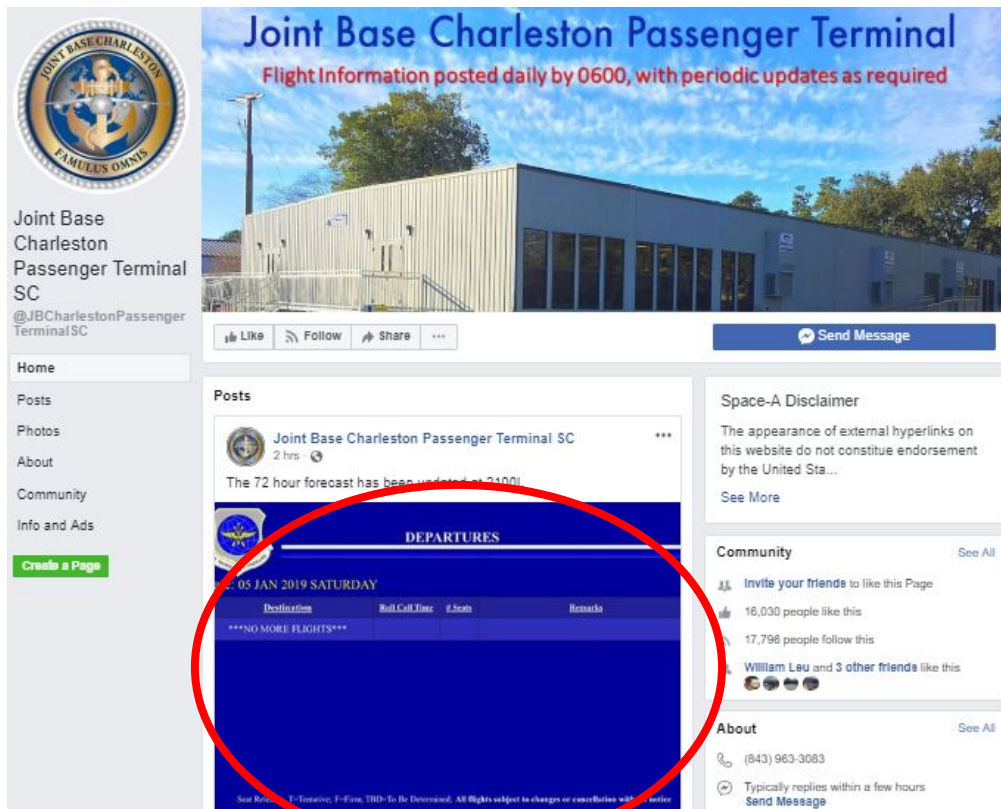
Seats [?]

Source [?]

04 May 2018


1510	Yokota AB, Japan	Travis AFB, California	N/A	✖
0730	Ramstein AB, Germany	JB Lewis-McChord, Washington	N/A	✖
1155	NS Rota, Spain	NS Norfolk, Virginia	N/A	✖
0610	JB Charleston, South Carolina	Colorado Springs, CO	N/A	✖
0625	JB Charleston, South Carolina	Alice Springs, Australia	N/A	✖
0715	Dover AFB, Delaware	Travis AFB, California	N/A	✖

# Step 0: Download Flight Schedule Photos



- Use the Facebook Graph API to download recent photos updated within 72 hrs.
  - Updated vs Posted because some terminals will continue to update a older posts with the new images. So the newly updated photo may look old by the post date.
- There are some certain heuristics needed to download all the right photos from all Space-A terminal pages. These are explained in the code comments.

# Step 1: Extract Text from Flight Schedule Photos



DEPARTURES FROM: NAVY OPERATED AMC TERMINAL NORFOLK, VA (NGU) 28 MAY 2018		
<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Fly Space-A was programmed to use Tesseract 3.05.02.
  - It is untested with the just released Tesseract 4, but the author expects better accuracy.
- The Tesseract config files used by the program are located under the *tesseract\_configfile* directory in the project folder.
- HTML Optical Character Recognition (HOOCR) is used to determine the position of text.

## Step 2: Determine Slide **Date**


- Find Month
- Find Day of Month and Year



**DEPARTURES FROM: NAVY OPERATED AMC  
TERMINAL NORFOLK, VA (NGU)  
28 MAY 2018**

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

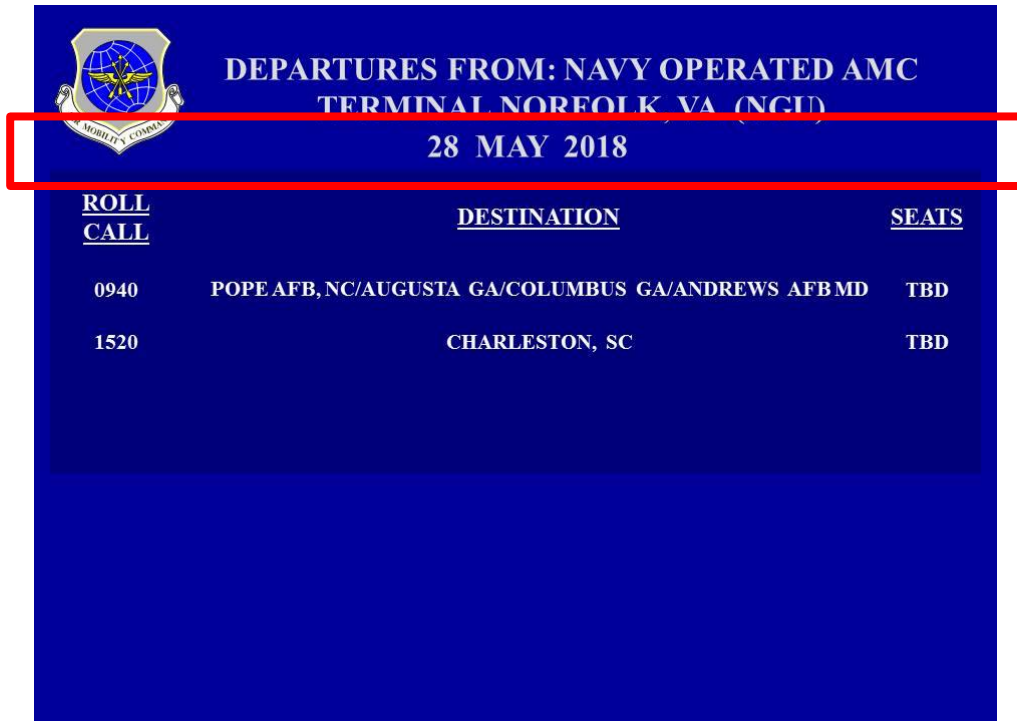
## Step 2a: Determine Slide Month



DEPARTURES FROM: NAVY OPERATED AMC TERMINAL NORFOLK, VA (NGU) 28 MAY 2018		
<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Generate full and abbreviated month strings (January, Jan, February, ...)
- Split entire slide OCR output text by a determined list of characters **space \n \r , : = ( ) . \* - /**
- Compare all strings after split
  - Compute levenshtein distance from the generated full and abbr month strings.
  - Pick the closest levenshtein distance match as the most likely Month.

## Step 2b: Determine Slide **Day of Month** and **Year**



<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Run regex to find Day of Month and Year on OCR from entire slide.
  - Observed formats
    - DDMMYY
    - DMMYYYY
- Get bounds of the found Month text using HOCR output from entire slide.
- Vertically crop the slide using the bounds to isolate the date line.  
**Highlighted**
- Re-run regex to find Day of Month and Year on the cropped slide.
- Pick the best date & sanity check if date is way off.
  - Whichever date is closest to current time.



# Step 3: Find area of destination listings

DEPARTURES FROM: NAVY OPERATED AMC  
TERMINAL NORFOLK, VA (NGU)  
28 MAY 2018

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

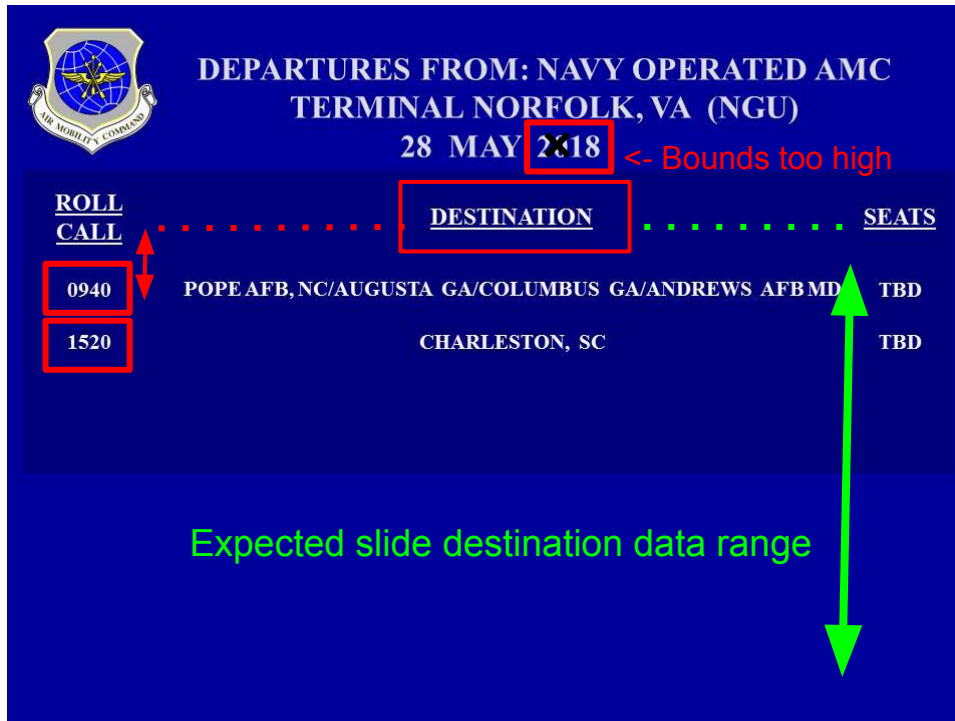
Bounds.y < 50% \* (total height)?

Expected slide destination data range

- Look for closest match of the keyword **DESTINATION** in OCR output from entire slide.
- Get bounds of the matched text using HOCR output from entire slide.
- **DESTINATION** is usually found near the top the slide.
- Check if **DESTINATION** bounds are within the top 50% the slide.
  - If bound are too low, we use the top of the slide as the bounds of **DESTINATION** -- basically not using this result to help isolate slide info.



## Step 4: Find flight Roll Call times



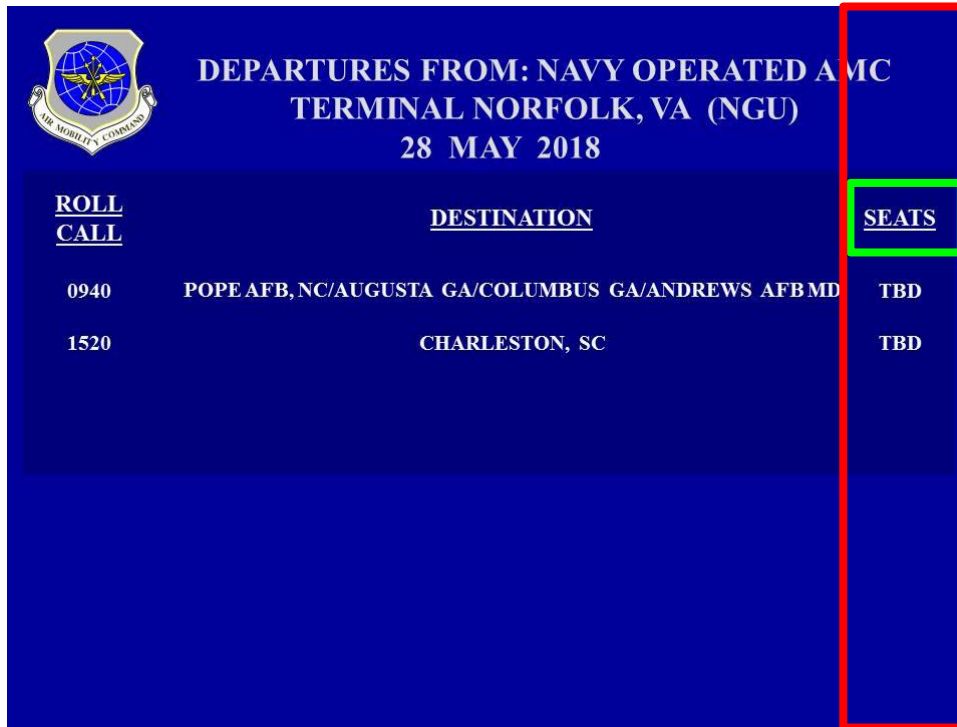
The image shows a flight departure slide with a blue background. At the top left is the Air Mobility Command logo. The title reads 'DEPARTURES FROM: NAVY OPERATED AMC TERMINAL NORFOLK, VA (NGU)'. Below the title is the date '28 MAY' followed by '2018', which is boxed in red. To the right of the date is a red annotation '<- Bounds too high'. Below the title is a table with three columns: 'ROLL CALL', 'DESTINATION', and 'SEATS'. The 'DESTINATION' column is boxed in red. The first row of the table has '0940' in the 'ROLL CALL' column, 'POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD' in the 'DESTINATION' column, and 'TBD' in the 'SEATS' column. The second row has '1520' in the 'ROLL CALL' column, 'CHARLESTON, SC' in the 'DESTINATION' column, and 'TBD' in the 'SEATS' column. A green double-headed arrow points from the 'DESTINATION' column to the text 'Expected slide destination data range' at the bottom. A red double-headed arrow points from the 'ROLL CALL' column to the '0940' and '1520' entries.

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

Expected slide destination data range

- Run regex on OCR output from entire slide looking for 24hr time listings (0000, 0900, 2359, ..)
- Get bounds of all found 24hr text.
- Check if the 24hr text is located below **DESTINATION** in the slide using our previously found **DESTINATION** bounds.
  - This conveniently filters out the date year if it is in YYYY format.

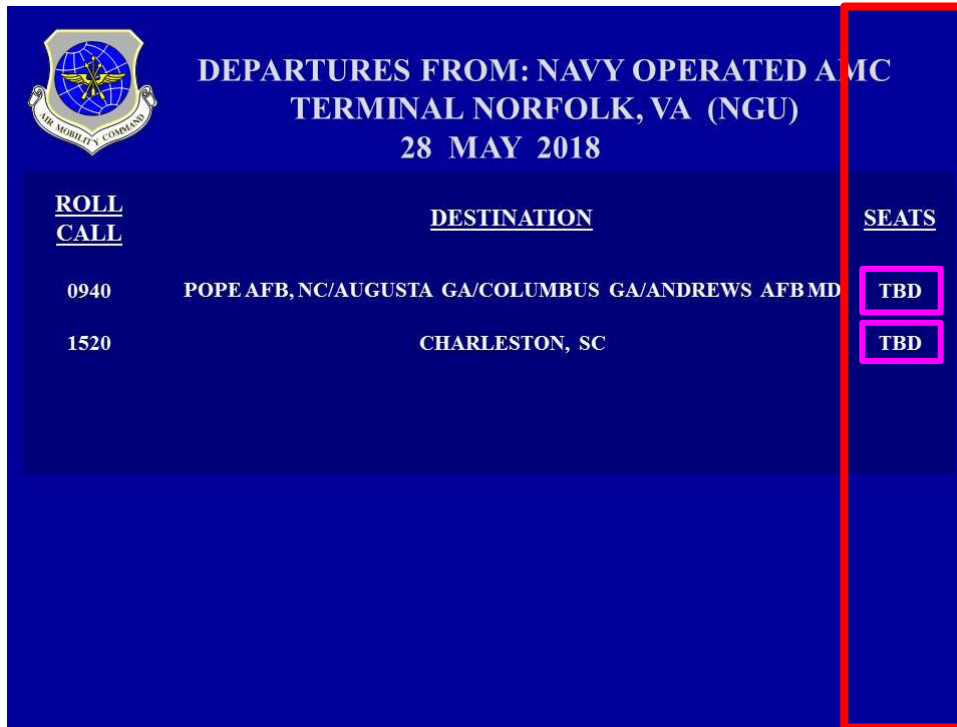
## Step 5: Find and isolate area of seats listings



<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Look for closest match of the keyword **SEATS** in OCR output from entire slide.
- Get **bounds** of the **SEATS** text using HOOCR output from entire slide.
- **Crop** slide horizontally to isolate seats listings.


## Step 6: Look for Seats listings



<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC/AUGUSTA GA/COLUMBUS GA/ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Run OCR on **cropped** slide showing seat listings.
  - Use a special determined character set **1234567890TBDFSP-** instead of A-Z, 0-9, etc.
  - Special character set includes only characters that are common in seat listings to reduce mismatches.
- Run **regex** on OCR of cropped slide showing seat listings to find seat info.
  - Common abbreviations: *F T SP TBD SP*
  - 023T = 23 tentative seats.
  - TBD = Seats to be determined.

# Step 7: Find Flight Destinations




**DEPARTURES FROM: NAVY OPERATED AMC  
TERMINAL NORFOLK, VA (NGU)  
28 MAY 2018**

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC / AUGUSTA GA / COLUMBUS GA / ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Load known destination list.
  - Premade list also includes multiple “match” words for destinations with multiple names.
  - *Baltimore-Washington International Airport -> BWI, etc*
  - Premade list also includes common words to NOT match due to mismatches: *FORT, PLEASE*
- **Find close matches** for destination names using OCR from entire slide.
  - Use levenshtein distance to determine closeness.

# Step 8: Find Flight Destinations




DEPARTURES FROM: NAVY OPERATED AMC  
TERMINAL NORFOLK, VA (NGU)  
28 MAY 2018

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC / AUGUSTA GA / COLUMBUS GA / ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD

- Load known destination list.
  - Premade list also includes multiple “match” words for destinations with multiple names.
  - *Baltimore-Washington International Airport -> BWI, etc*
  - Premade list also includes common words to NOT match due to mismatches: *FORT, PLEASE*
- **Find close matches** for destination names using OCR from entire slide.
  - Use levenshtein distance to determine closeness.

# Step 9: Match it all up

- Using two examples to explain matching due to the format.



**DEPARTURES FROM: NAVY OPERATED AMC  
TERMINAL NORFOLK, VA (NGU)  
28 MAY 2018**

<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0940	POPE AFB, NC / AUGUSTA GA / COLUMBUS GA / ANDREWS AFB MD	TBD
1520	CHARLESTON, SC	TBD



**DEPARTURES  
JOINT BASE PEARL HARBOR-HICKAM  
SUNDAY, 27 MAY 2018**

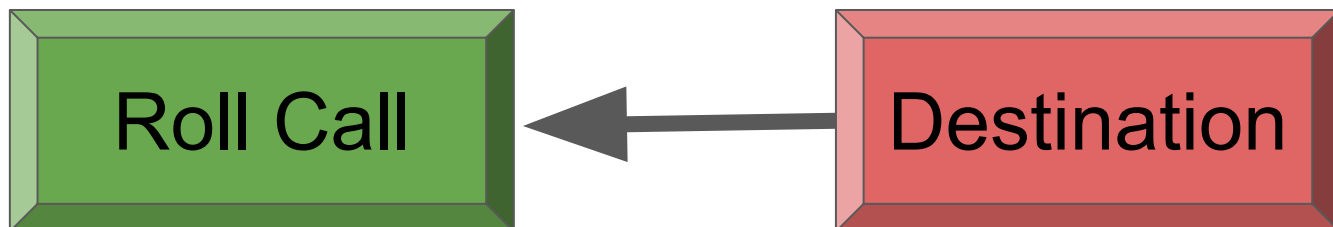
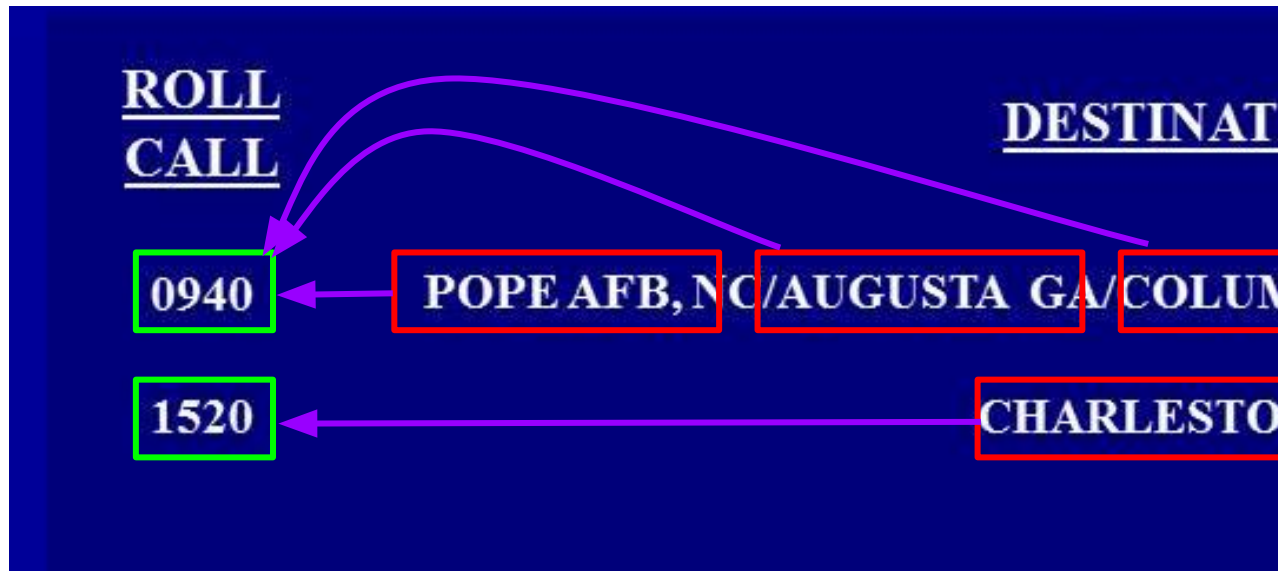
<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0555	TRAVIS AFB, CA	SP
0911	MISAWA AB, JAPAN <small>*SOFA STAMP REQUIRED*</small>	4F
	<small>*THRU-LOAD PASSENGERS ARRIVE 1 HOUR PRIOR*</small>	
1723	MARCH AFB, CA TRAVIS AFB, CA	SP
1925	ALICE SPRINGS, AUSTRALIA ANDERSEN AFB, GUAM RICHMOND, AUSTRALIA	SP

Destinations listed in alphabetical order  
Seat Releases: **T**: Tentative **F**: Firm **SP**: Seats Pending  
ALL FLIGHTS ARE SUBJECT TO CHANGE WITHOUT NOTICE

# Step 9a: Match Roll Calls to Destinations

- Link Roll Call with Destination on same horizontal axis

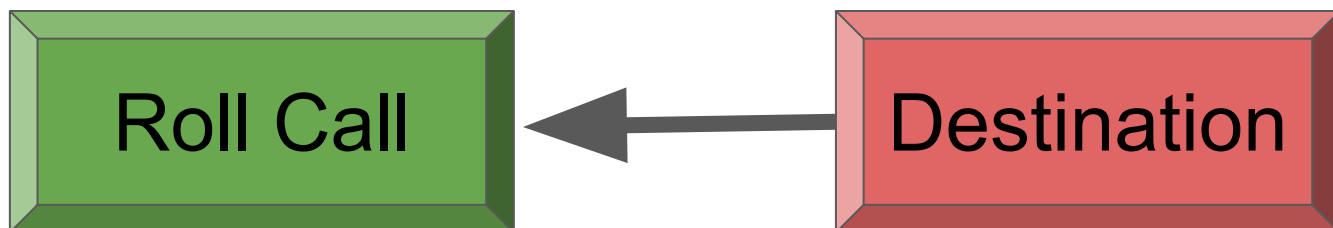
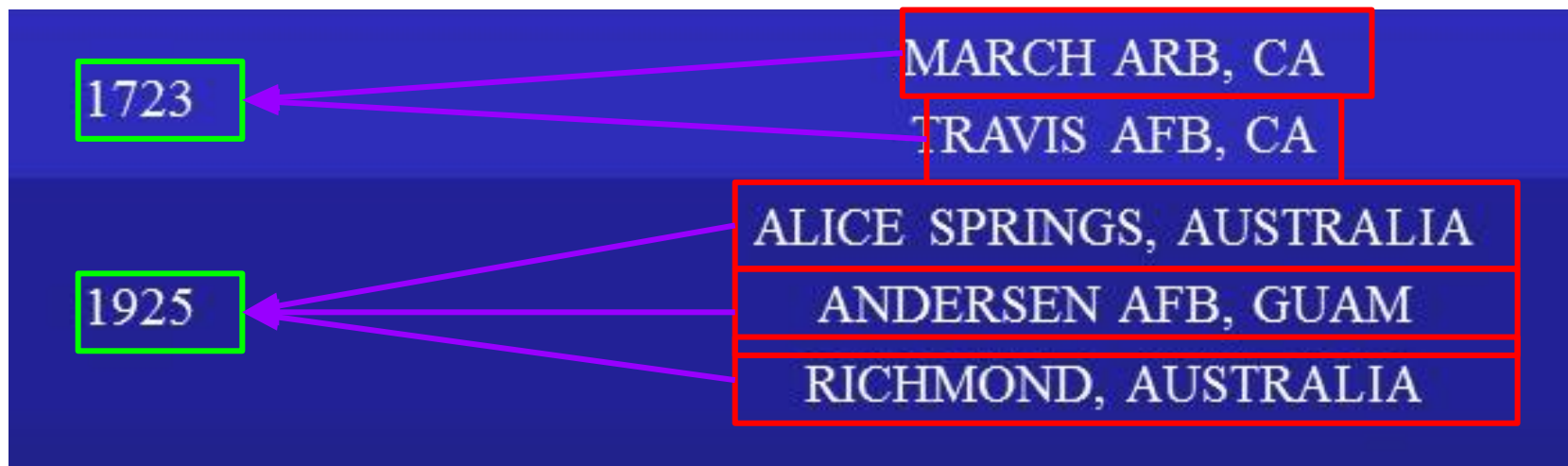
```
linkRollCallsToNearestDestinations(rcs []RollCall,  
destsArray []Destination)
```





# Why link from Destination to Roll Call for now?

- Store a reference to the to Roll Call in the Destination object.
- Multiple destinations for one roll call.
- A many-to-one relationship between each flight destination found and a roll call time.

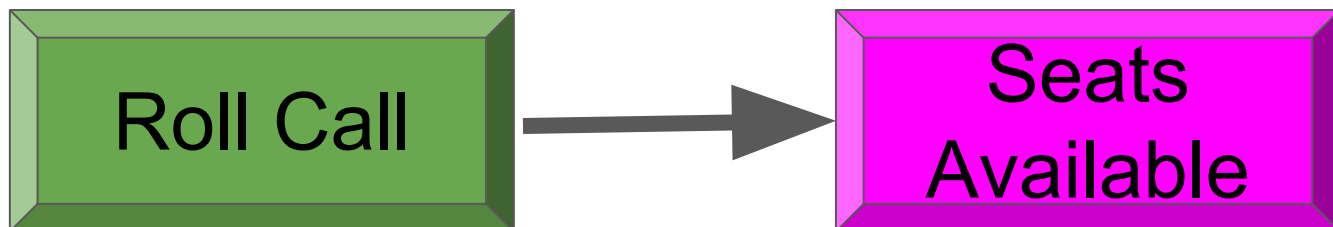


## Step 9b: Link Roll Calls to Seats Available

- Link Roll Call with the Seats Available on the same horizontal axis

```
linkRollCallsToNearestSeatsAvailable(rcs [ ]RollCall,  
saArray [ ]SeatsAvailable)
```

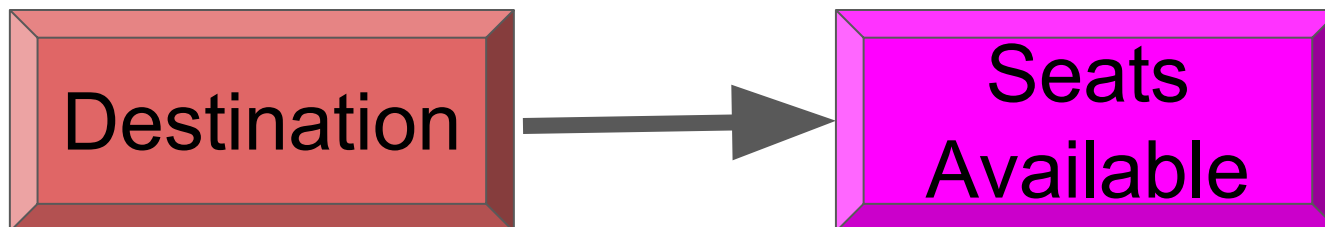
<u>ROLL CALL</u>	<u>DESTINATION</u>	<u>SEATS</u>
0555	TRAVIS AFB, CA	SP
0911	MISAWA AB, JAPAN *SOFA STAMP REQUIRED*	4F
*THRU-LOAD PASSENGERS ARRIVE 1 HOUR PRIOR*		



## Step 9c: Link Destinations to Seats Available

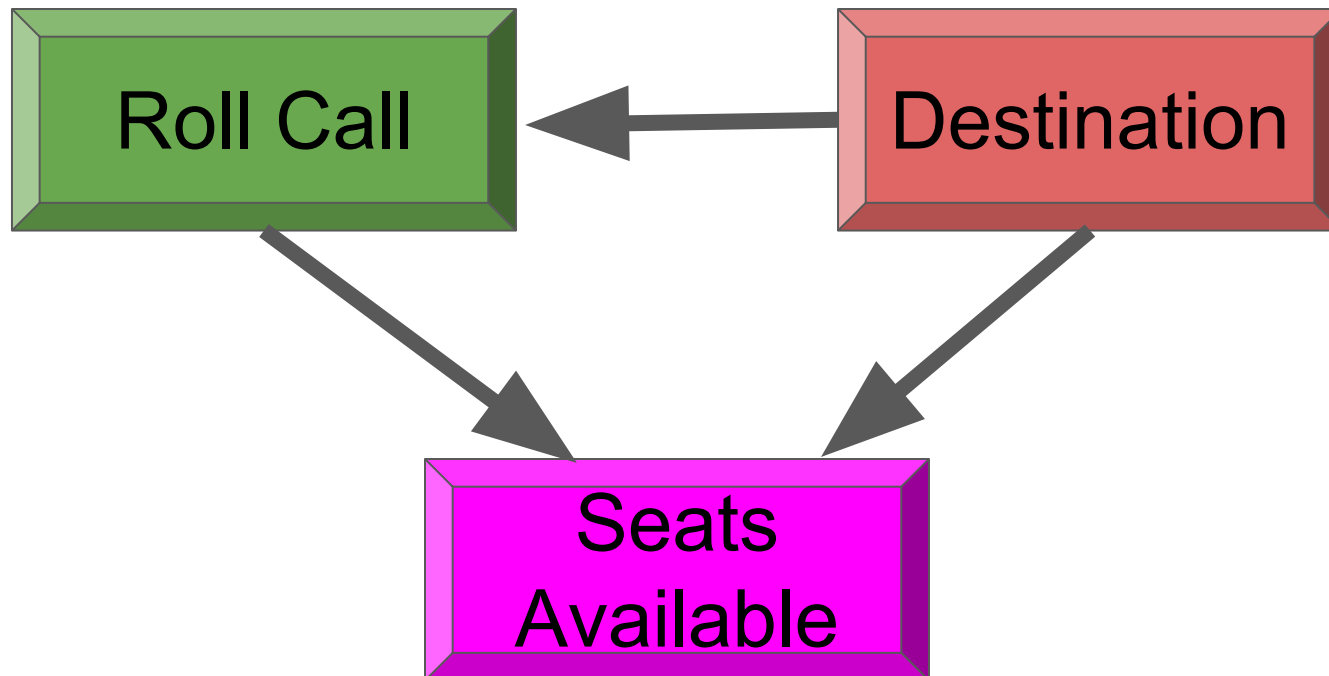
- Link Destinations with the Seats Available on the same horizontal axis

```
linkDestinationsToNearestSeatsAvailable(dests  
[]Destination, saArray []SeatsAvailable)
```

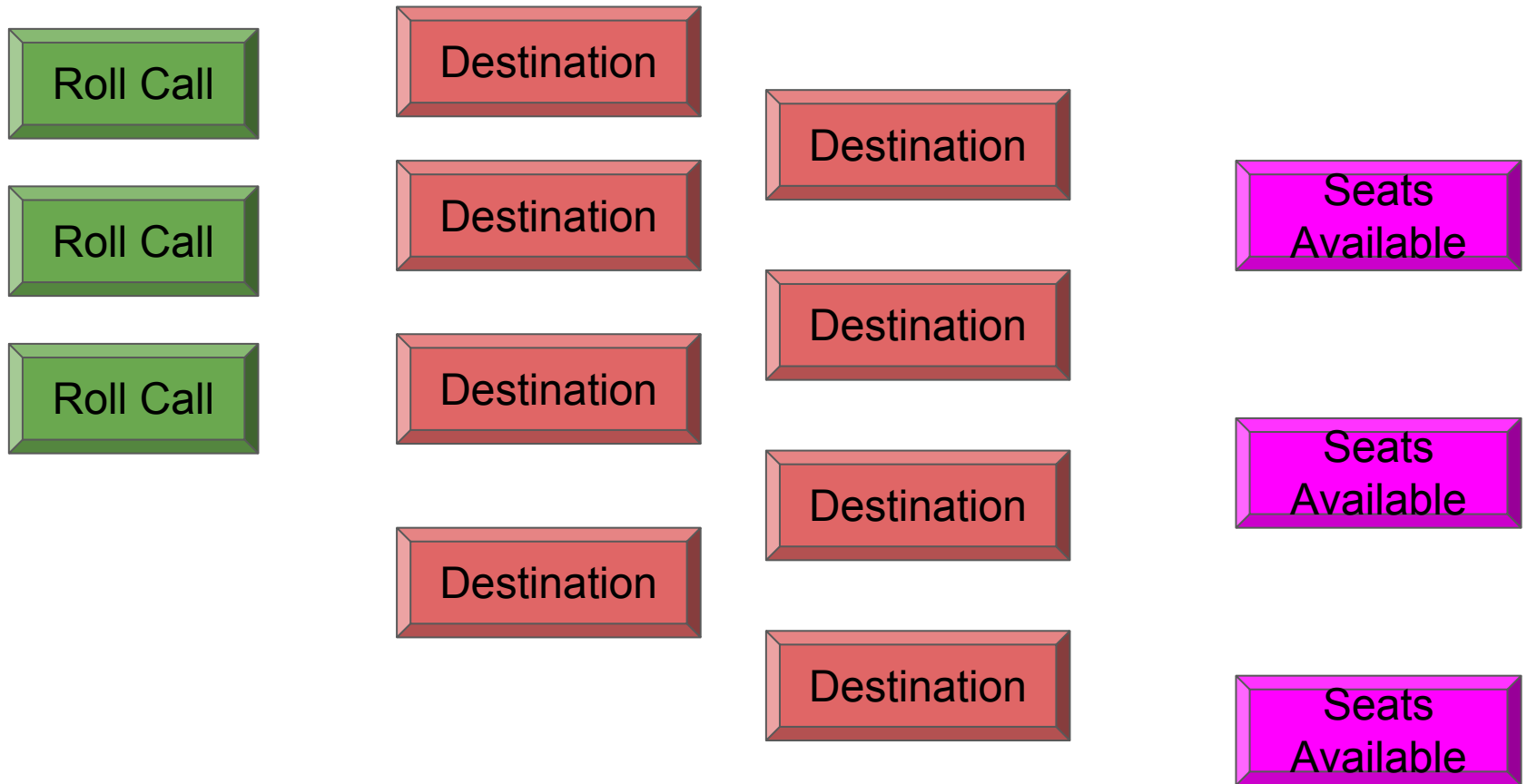
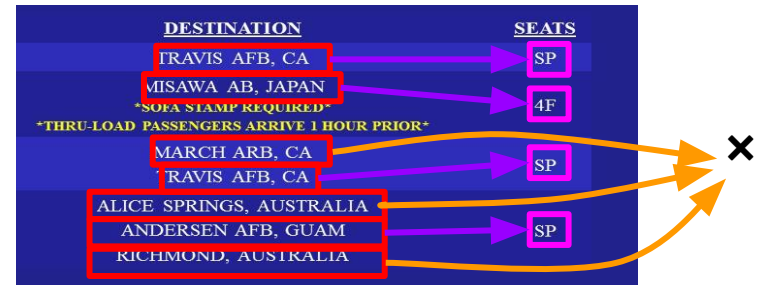


# Why link from Roll Call / Destination to Seats Available?

- Usually a one-to-to relationship between a roll call and seats available.
  - Sometimes the Space-A terminal will try to put multiple flights for a roll call which we will not handle.
- Sometimes a many-to-one relationship between each flight destination found and a seats available listing.



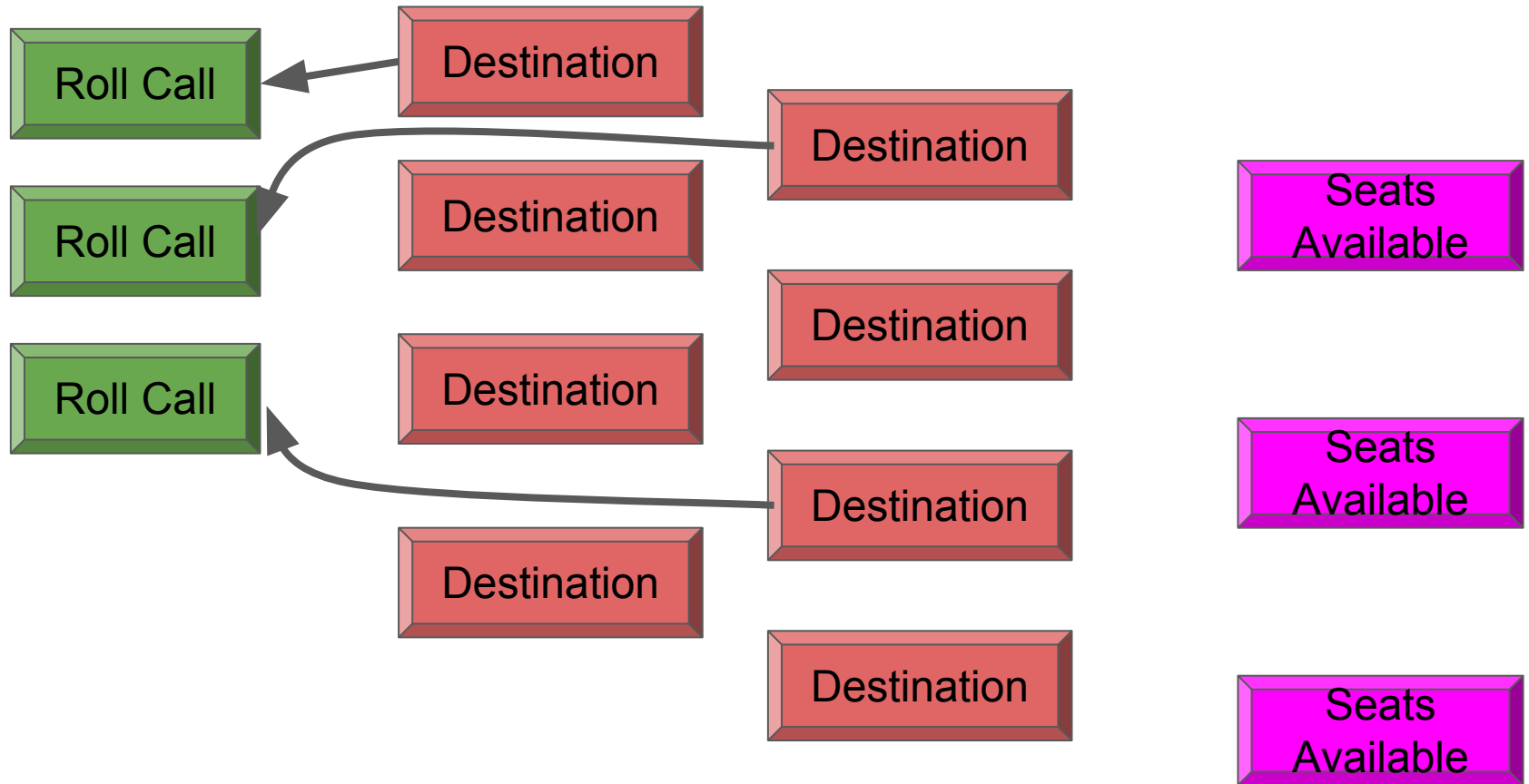
# Rough Overview of Object References at this point



# Rough Overview of Links (Object References)

DESTINATION	SEATS
TRAVIS AFB, CA	SP
MISAWA AB, JAPAN	4F
*SOFA STAMP REQUIRED*	
MARCH ARB, CA	SP
TRAVIS AFB, CA	SP
ALICE SPRINGS, AUSTRALIA	SP
ANDERSEN AFB, GUAM	SP
RICHMOND, AUSTRALIA	SP

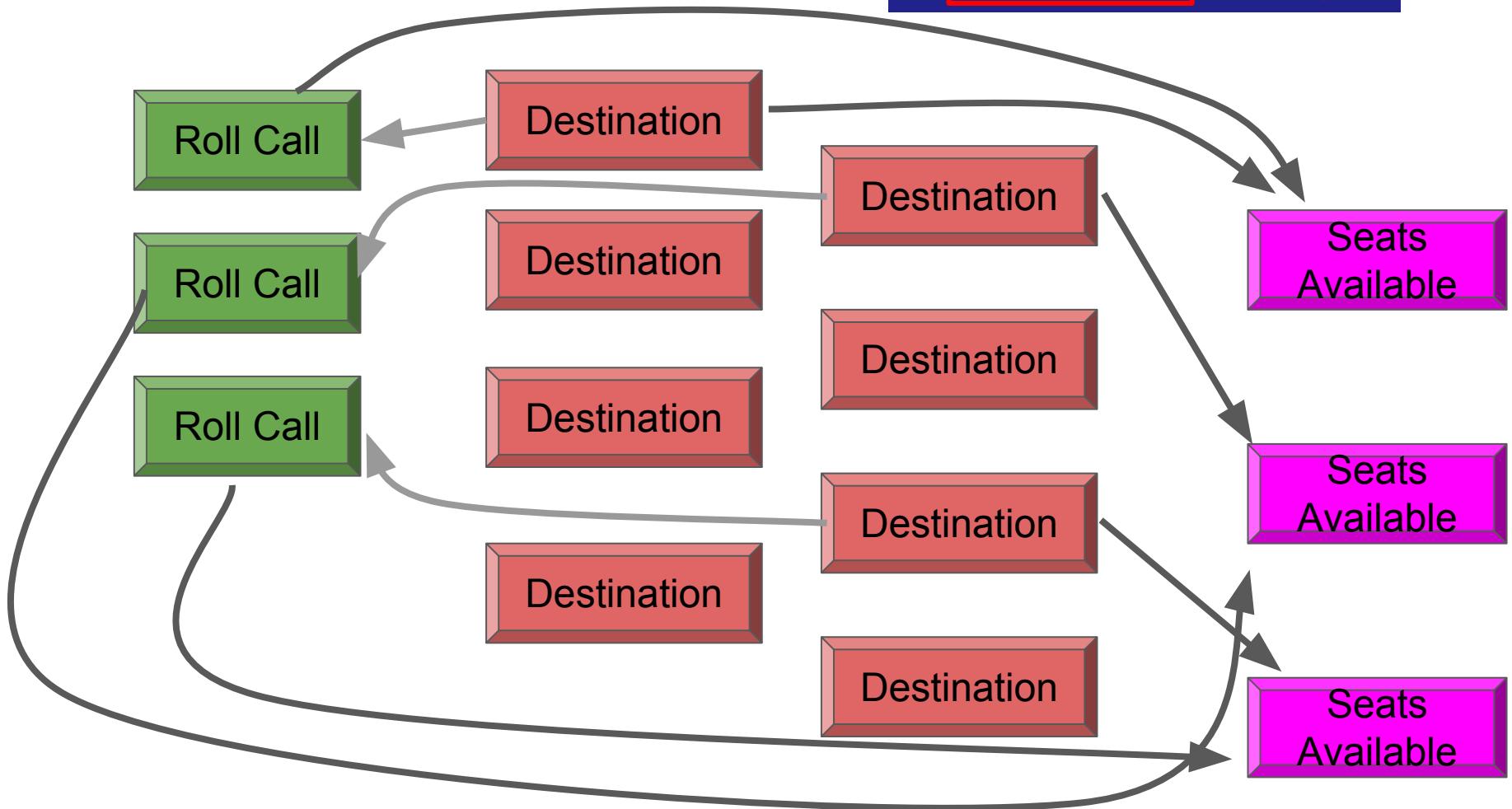
\*THRU-LOAD PASSENGERS ARRIVE 1 HOUR PRIOR\*



# Rough Overview of Links at this point

DESTINATION	SEATS
TRAVIS AFB, CA	SP
MISAWA AB, JAPAN	4F
*SOFA STAMP REQUIRED*	
MARCH ARB, CA	SP
TRAVIS AFB, CA	SP
ALICE SPRINGS, AUSTRALIA	SP
ANDERSEN AFB, GUAM	SP
RICHMOND, AUSTRALIA	SP

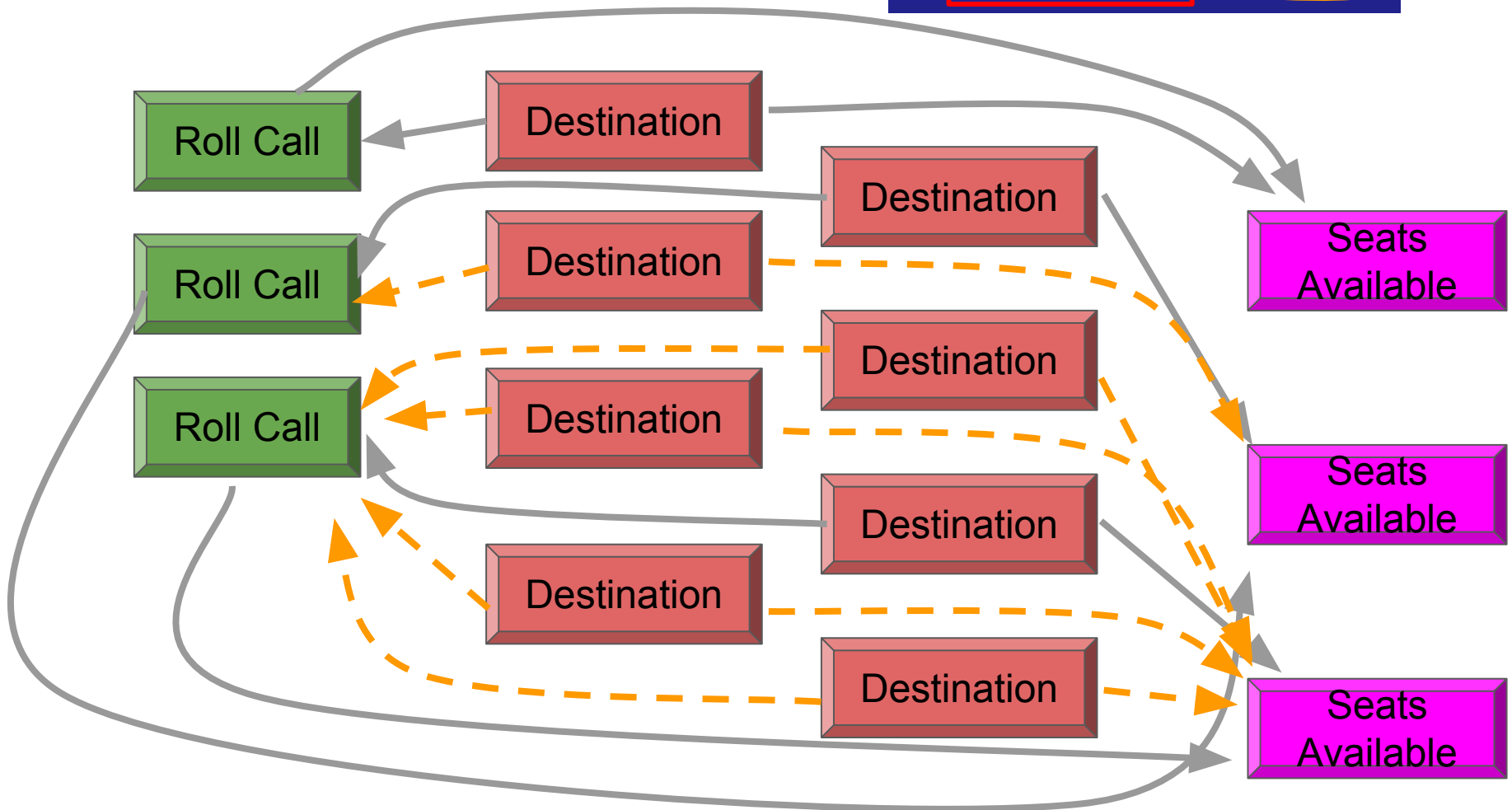
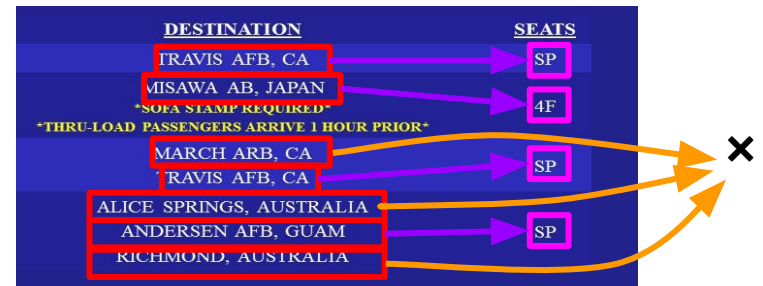
Orange arrows from the table point to a large 'X' on the right, indicating a conflict or error state.





# Links we need to figure out

DESTINATION	SEATS
TRAVIS AFB, CA	SP
MISAWA AB, JAPAN	4F
*SOFA STAMP REQUIRED*	
MARCH AFB, CA	SP
TRAVIS AFB, CA	SP
ALICE SPRINGS, AUSTRALIA	SP
ANDERSEN AFB, GUAM	SP
RICHMOND, AUSTRALIA	SP

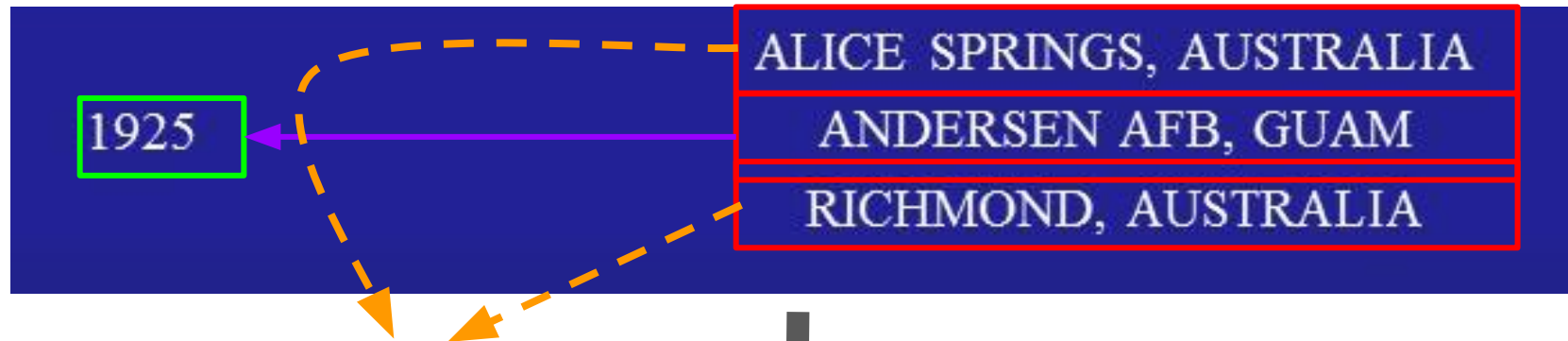


## Single **Flight** Representation

```
type Flight struct {  
    Origin      string  
    Destination string  
    RollCall    time.Time  
    UnknownRollCallDate bool  
    SeatCount   int  
    SeatType    string  
    Cancelled   bool  
    PhotoSource string  
    SourceDate  time.Time  
}
```

1 Origin - 1 Destination - 1 Roll Call

How to reconcile our many-to-one relationships into one-to-one relationships? What if we don't know they are related yet?



No match.  
Roll Call is not on same horizontal axis.

1925	ALICE SPRINGS
1925	ANDERSEN AFB
1925	RICHMOND

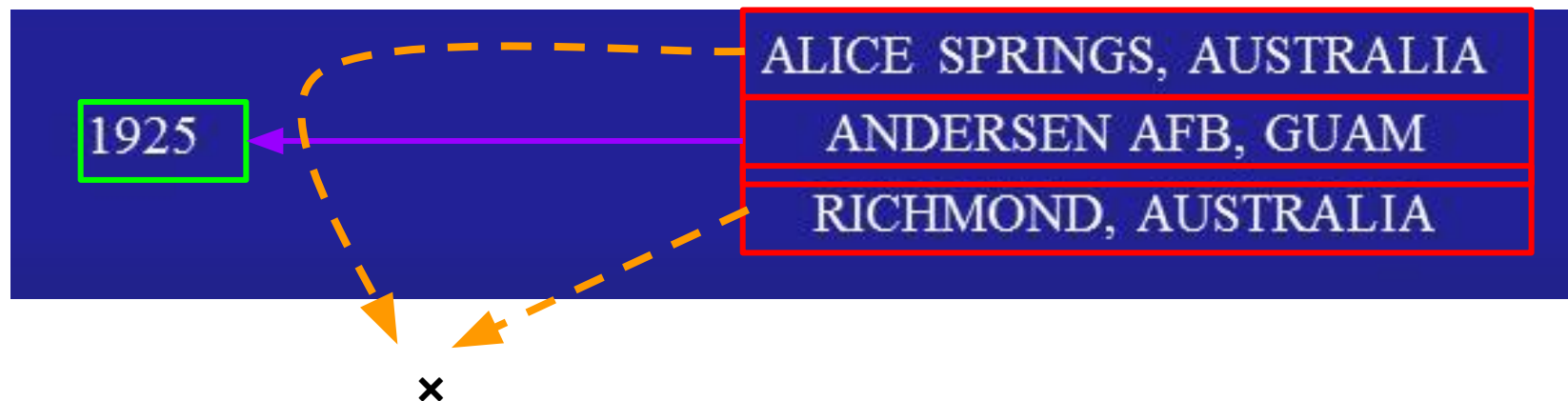
# “Group” representation

- A **group** holds multiple Destinations and a single related *RollCall*
- Seats Available is not linked from **group** because the *RollCall* will have a reference to it.
- A linked *RollCall* indicates that this **group** is all matched up with a valid *RollCall*. We will call the initial *Destination* that is already matched up with a *RollCall* an **anchor Destination**.

```
// "Grouping" of multiples Destinations for single RollCall/SeatsAvailable
type Grouping struct {
    Destinations []Destination

    // non nil value indicates Grouping contains 'anchor' Destination
    LinkedRollCall *RollCall
    SharedInfo
}
```

# Example Groups



Group #	Destinations	LinkedRollCall
1	ALICE SPRINGS	
2	ANDERSEN AFB	1925
3	RICHMOND	


Anchor Group!  
Because it is already  
matched with a RollCall  
at this point based on its  
physical position in the  
slide.

## Step 10: Grow by Absorbing Groups

`combineDestinationGroupsToAnchorDestinations(groupsP *[]Grouping)`

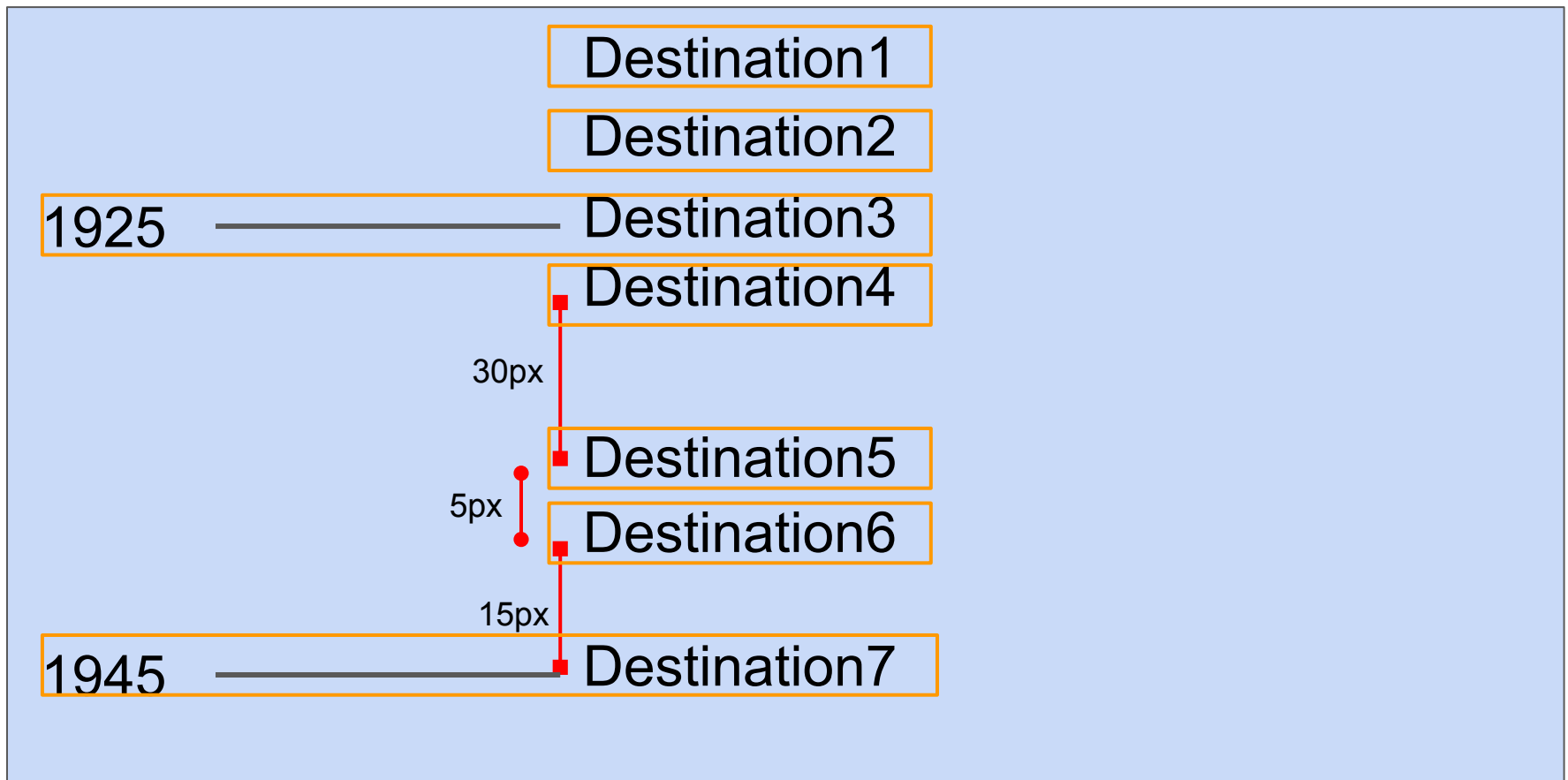
- Create a Group for each Destination and related RollCall. Otherwise just make a Group with only the single Destination and no linked RollCall.
- If a Group is NOT an Anchor Group:
  - Look for the physically closest Group.
  - Combine the closest Group by absorbing it.
    - Copy all the closest Group's Destinations and linked RollCall.
  - Repeat until this Group becomes an Anchor Group by absorbing another Group with a linked RollCall.
- Move onto the next group.
- Repeat until all Groups are Anchor Groups.

# Grow by Absorbing Groups (Exaggerated Example)

 = indicates a Group (in memory, no relation to physical location on the slide)

## Roll Calls

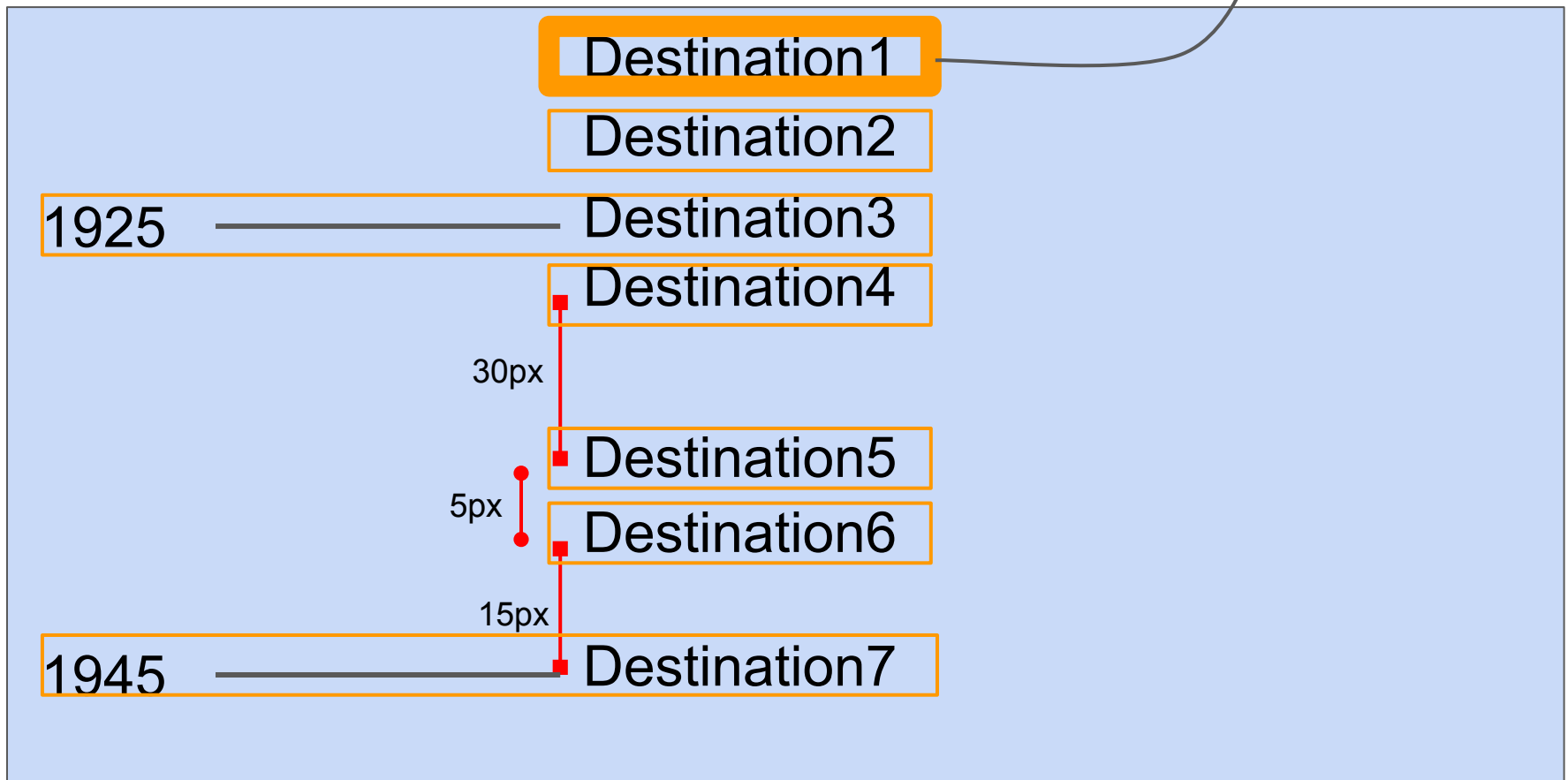
## Destinations





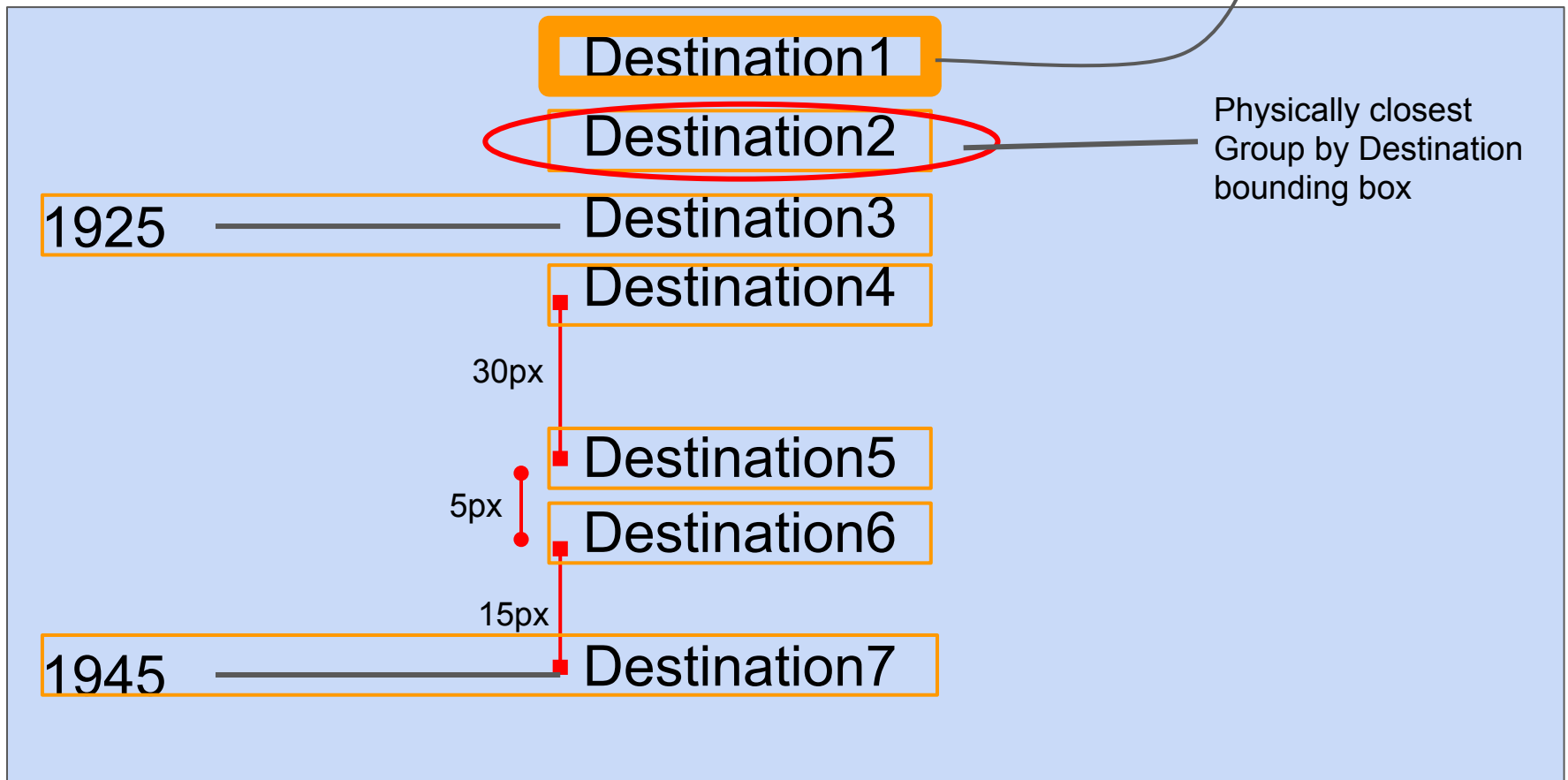
# Grow by Absorbing Groups (Exaggerated Example)

- If a Group is NOT an Anchor Group:
  - Look for the physically closest Group. Growing this group



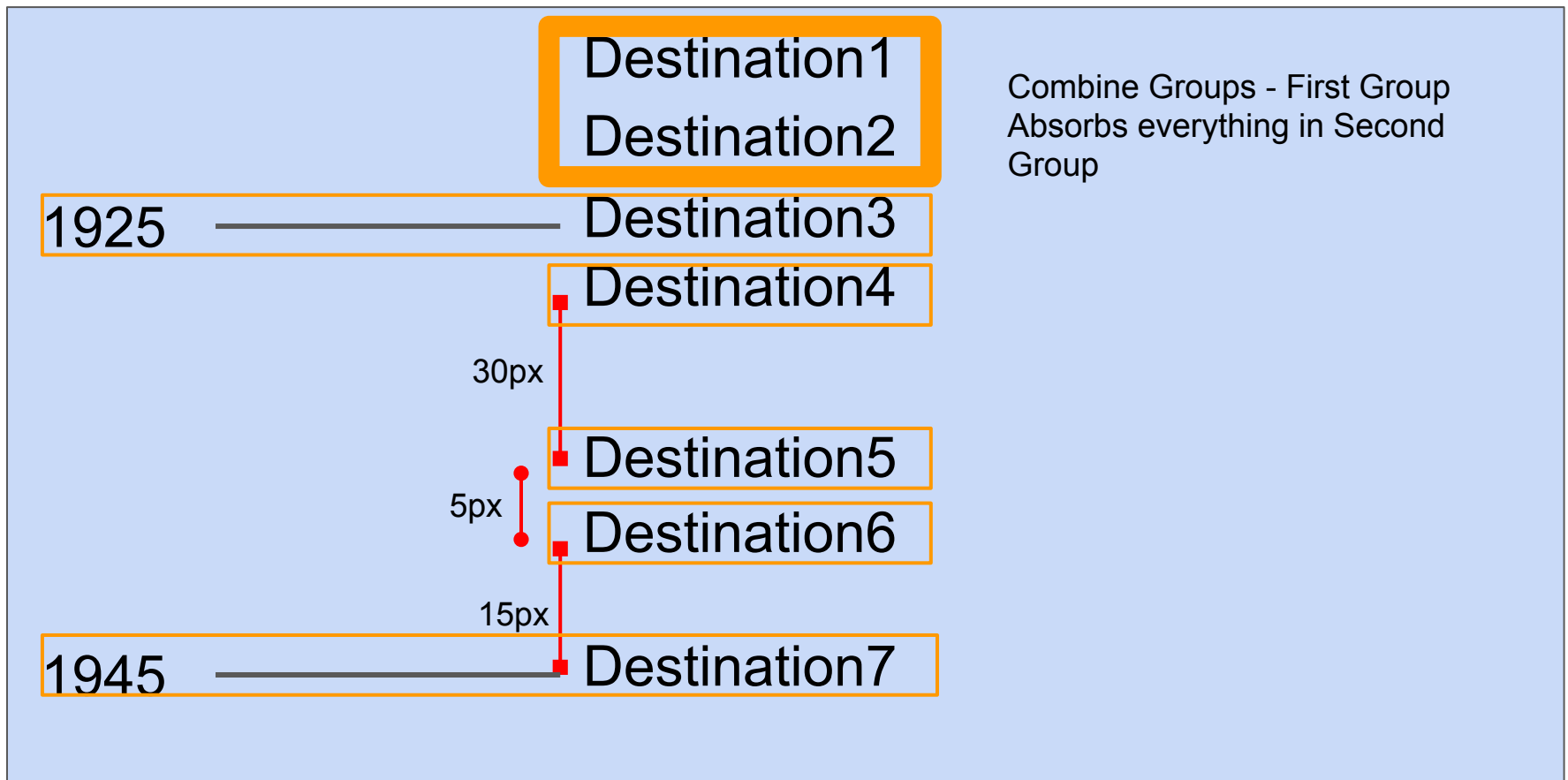
# Grow by Absorbing Groups (Exaggerated Example)

- If a Group is NOT an Anchor Group:
  - Look for the physically closest Group. Growing this group



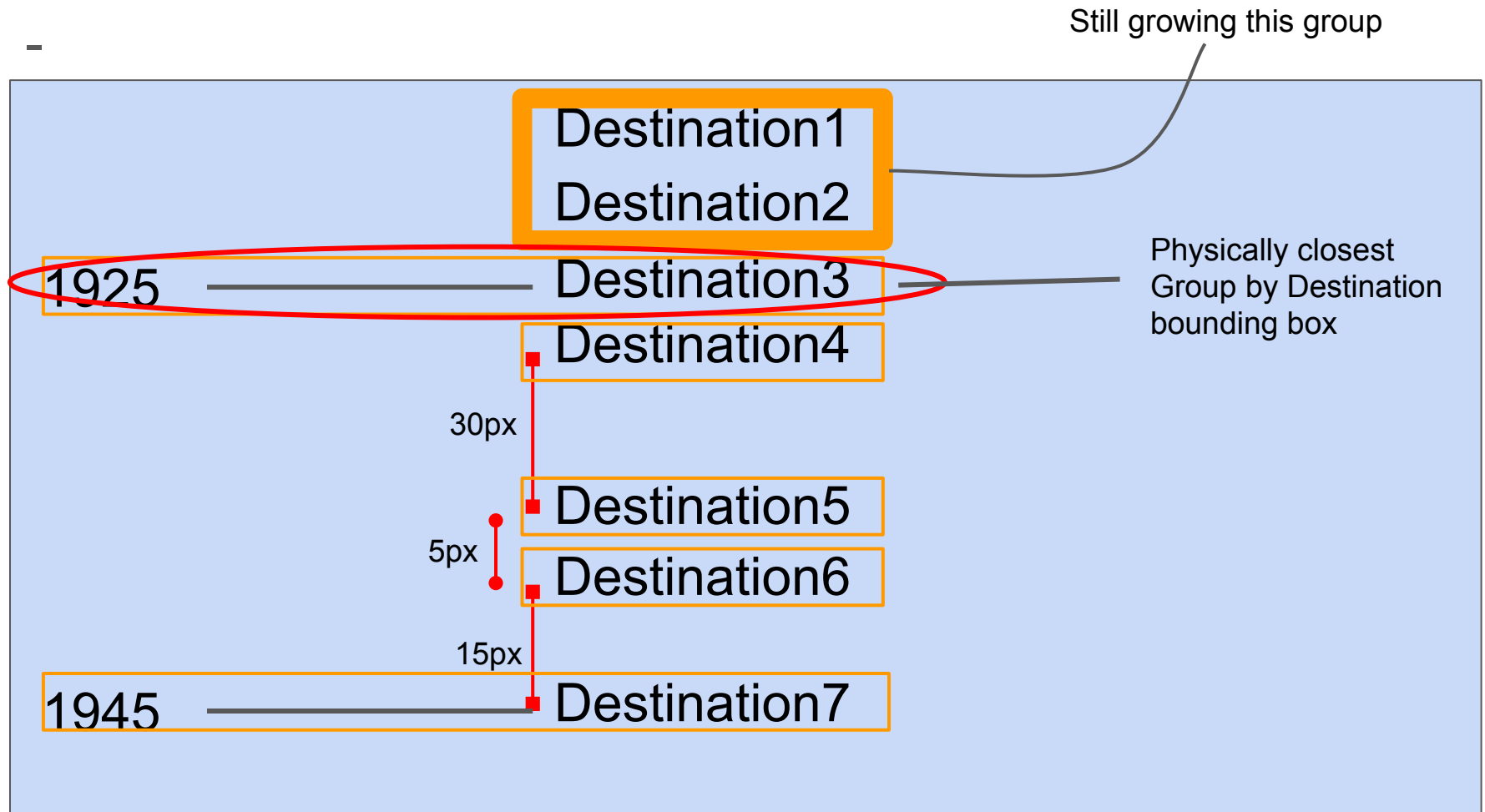
# Grow by Absorbing Groups (Exaggerated Example)

- Combine the closest Group by absorbing it.
  - Copy all the closest Group's Destinations and linked RollCall.



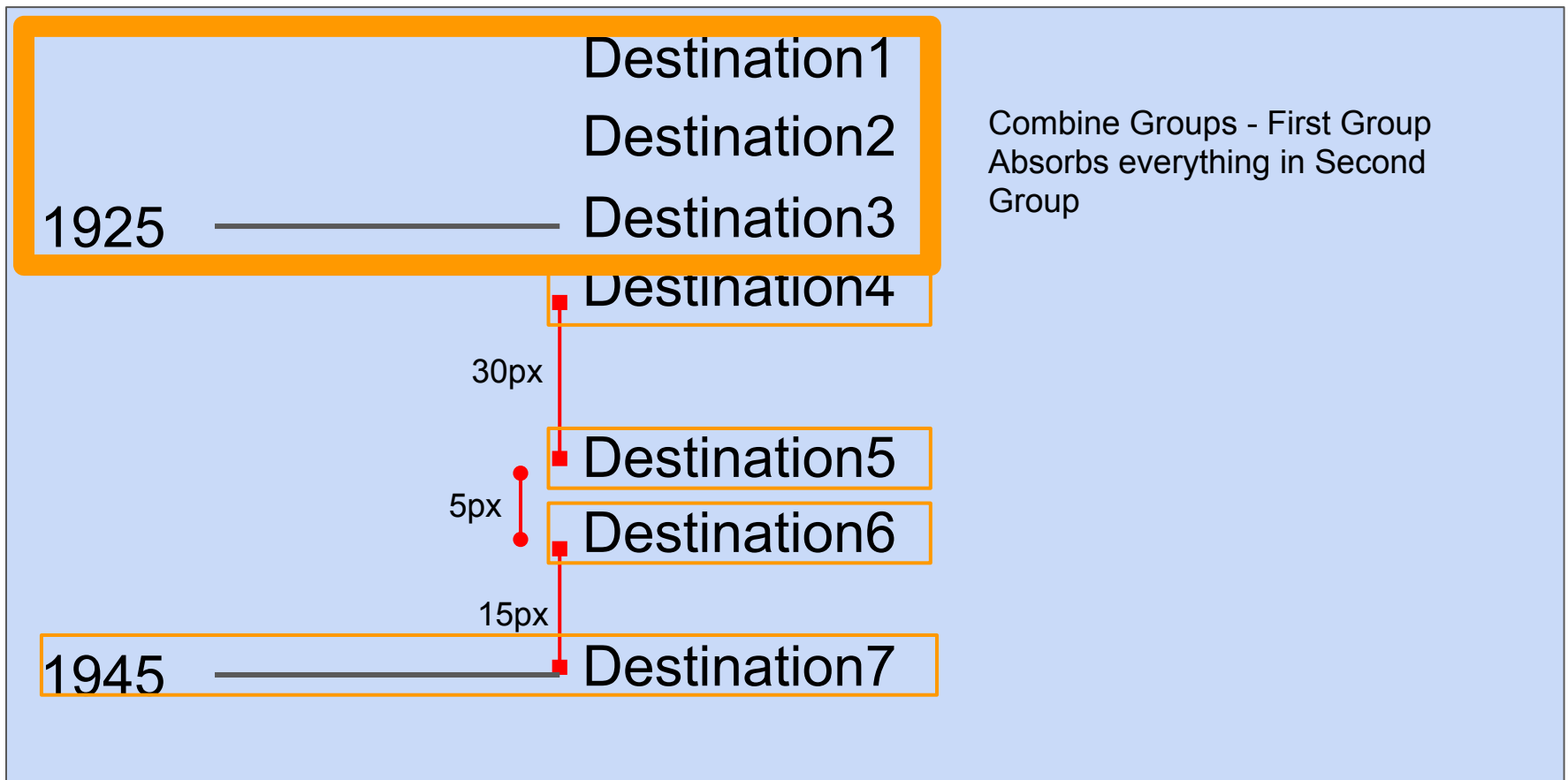
# Grow by Absorbing Groups (Exaggerated Example)

- If a Group is NOT an Anchor Group:
  - Look for the physically closest Group.



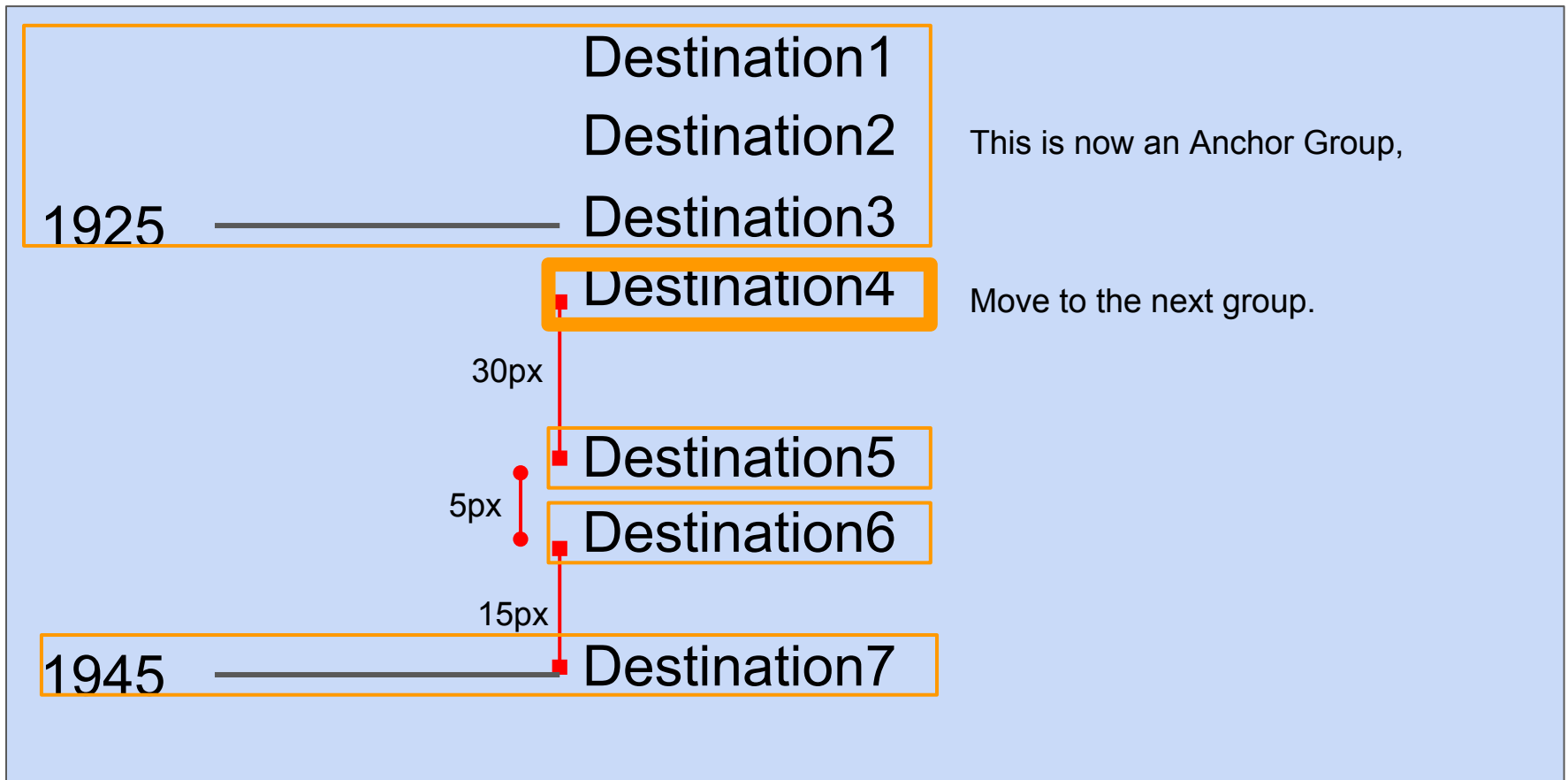
# Grow by Absorbing Groups (Exaggerated Example)

- Combine the closest Group by absorbing it.
  - Copy all the closest Group's Destinations and linked RollCall.



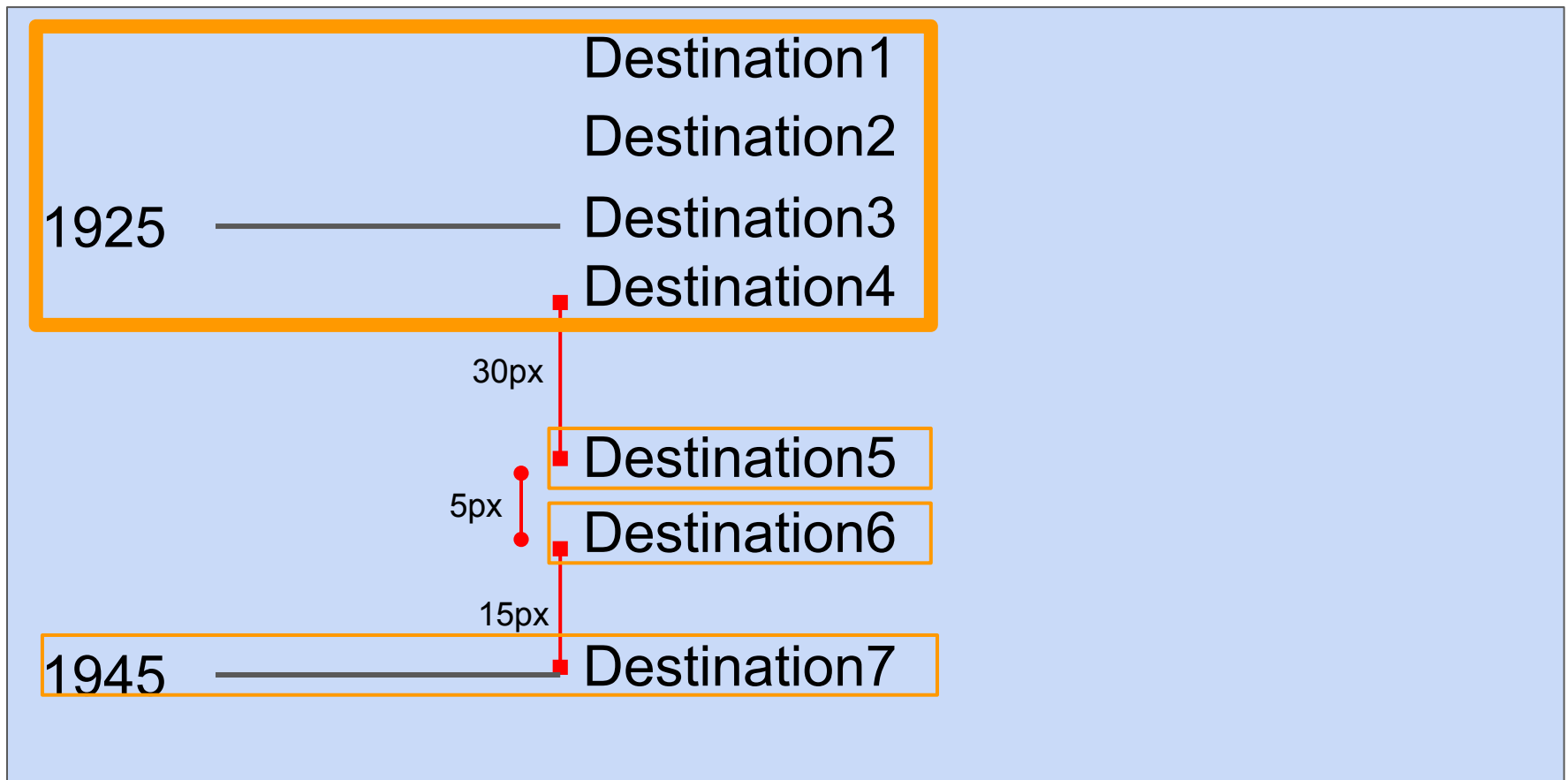
# Grow by Absorbing Groups (Exaggerated Example)

- Repeat until this Group becomes an Anchor Group by absorbing another Group with a linked RollCall.
- Move onto the next group.



# Grow by Absorbing Groups (Exaggerated Example)

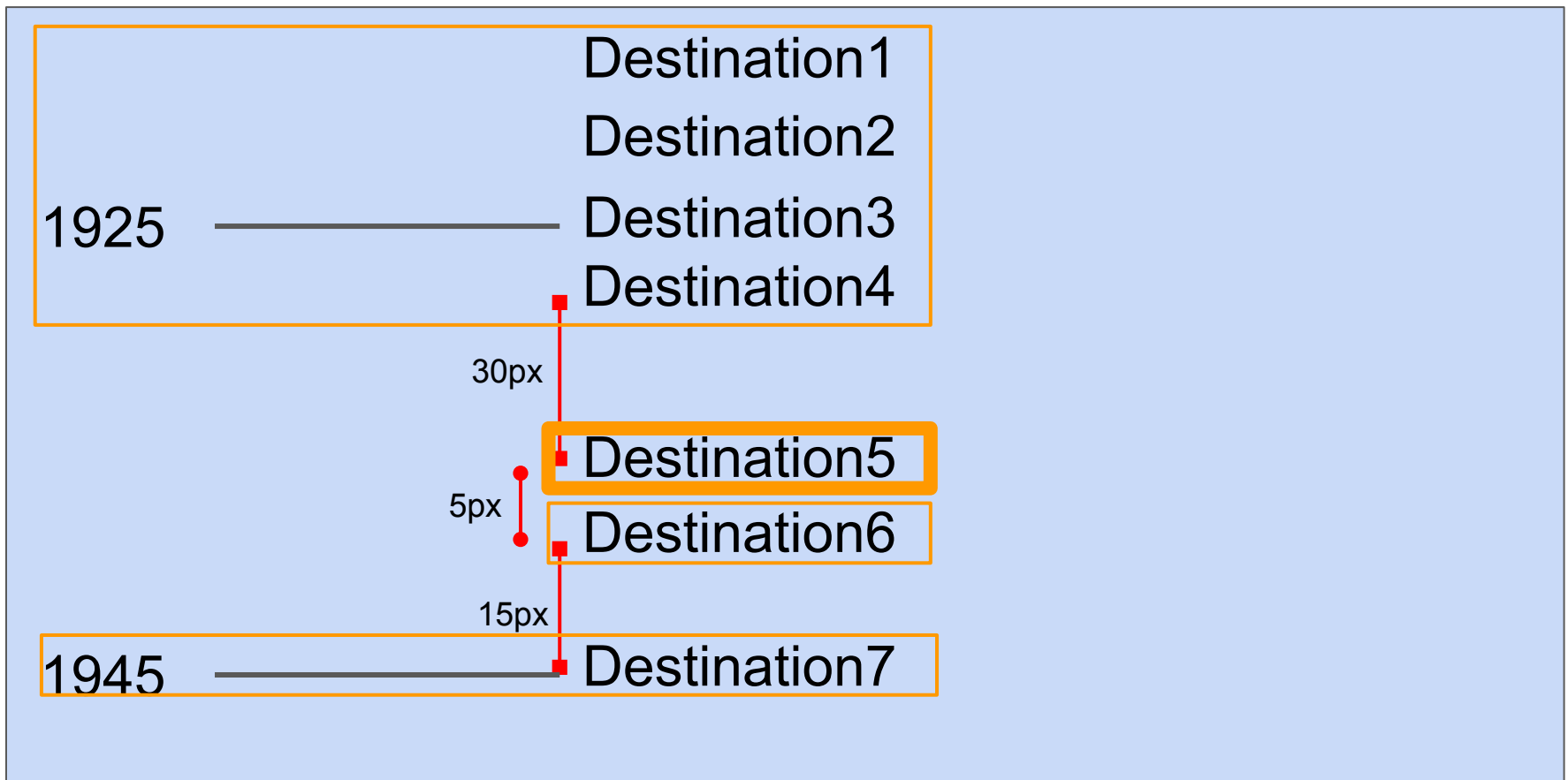
- Repeat until all Groups are Anchor Groups





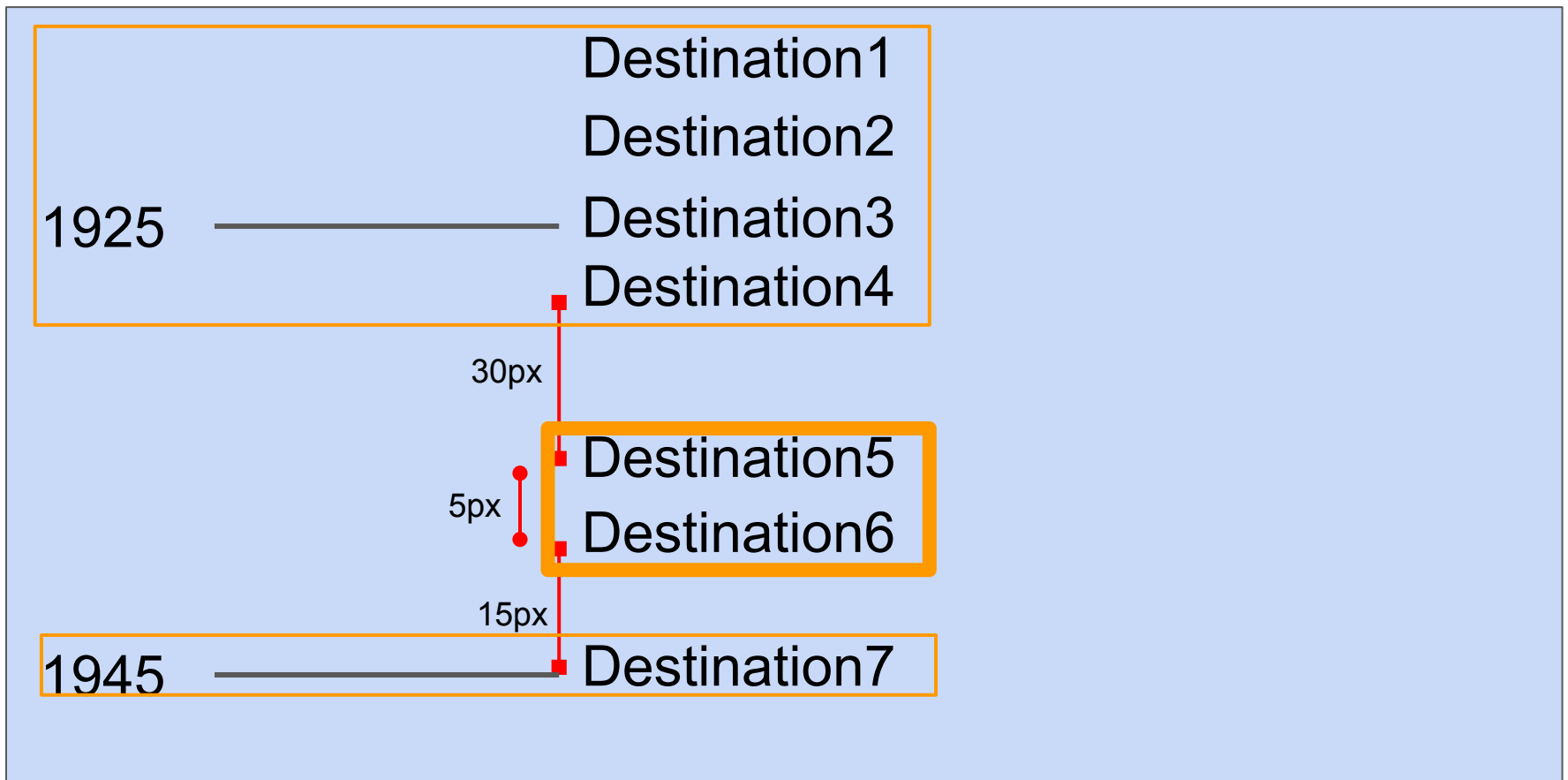
# Grow by Absorbing Groups (Exaggerated Example)

- Repeat until all Groups are Anchor Groups



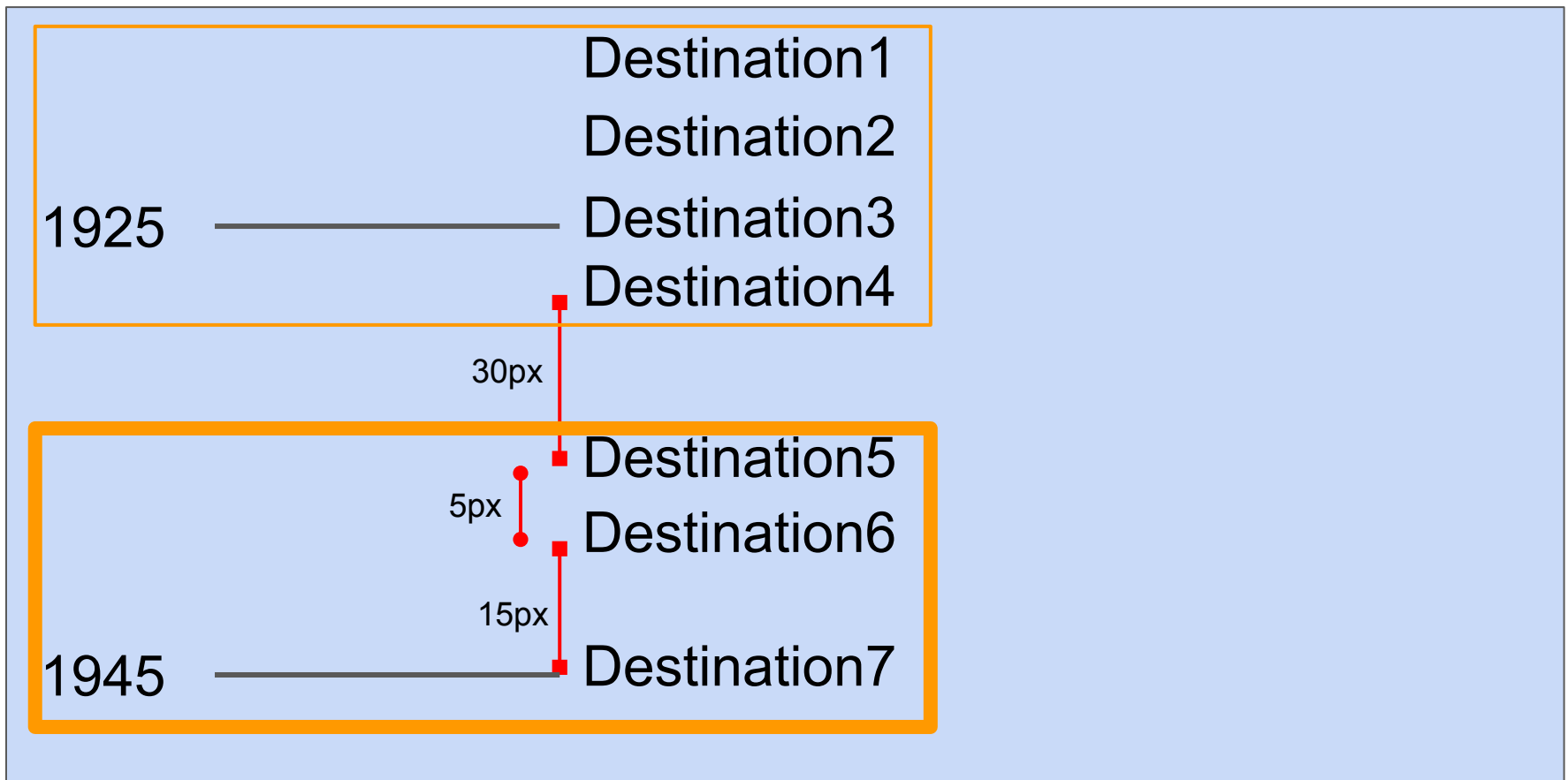
# Grow by Absorbing Groups (Exaggerated Example)

- Repeat until all Groups are Anchor Groups



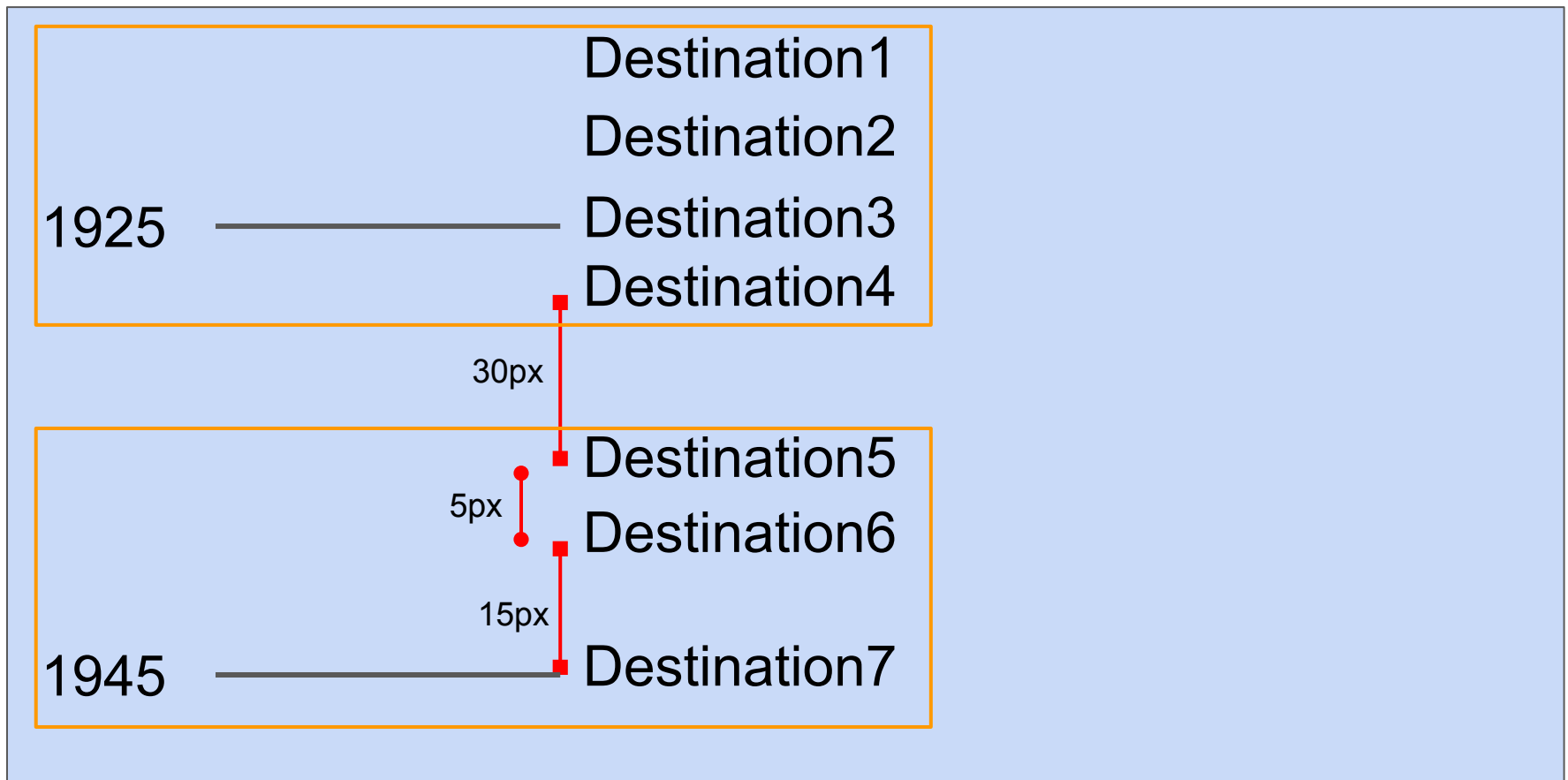
# Grow by Absorbing Groups (Exaggerated Example)

- Repeat until all Groups are Anchor Groups



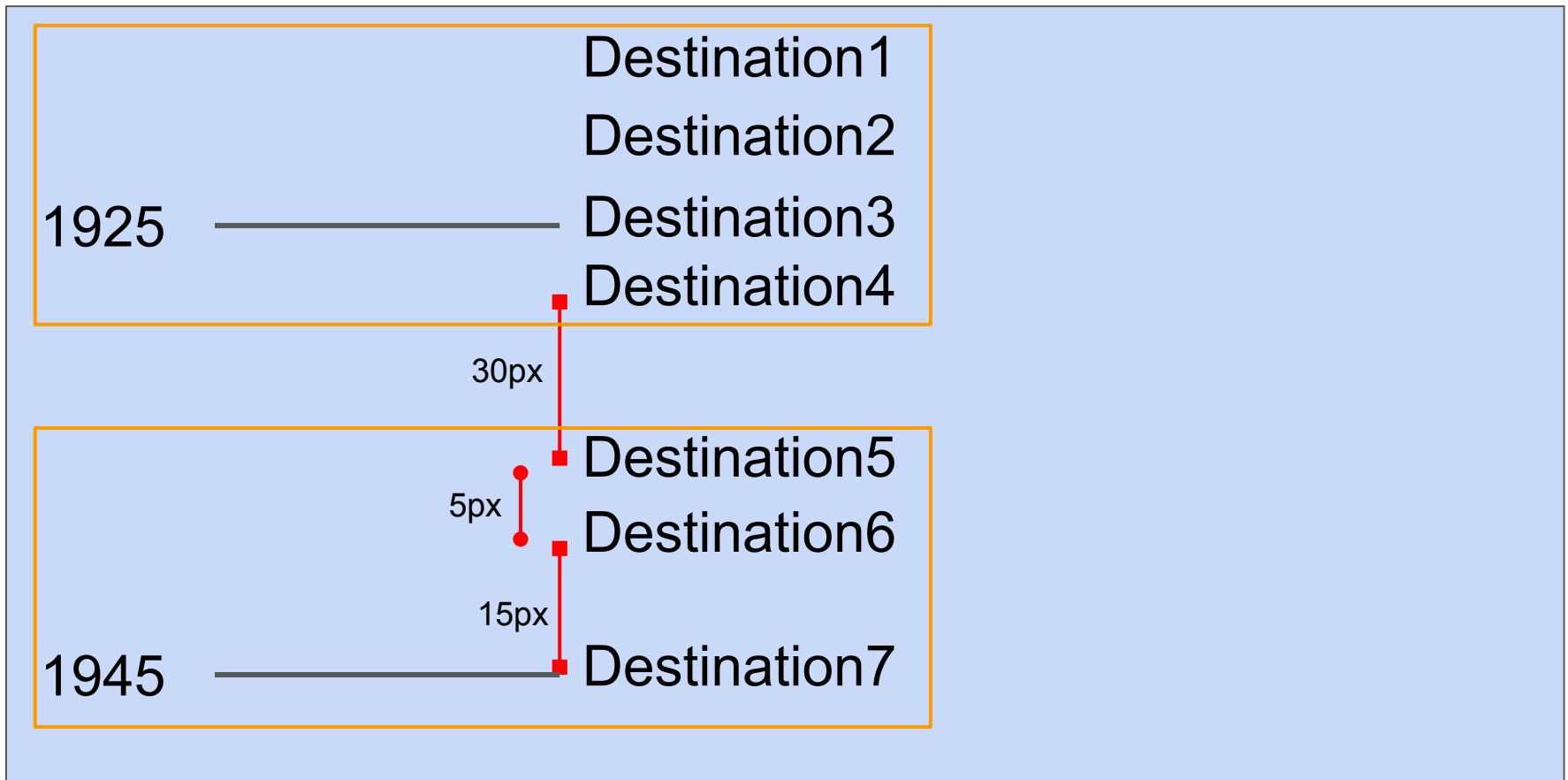
# Grow by Absorbing Groups (Exaggerated Example)

- All Groups are now Anchor Groups
- Completed Combining Groups



# Create a Flights from all Groups

- All Destinations are associated with a RollCall
- Any associated RollCall and Seats Available should be linked previously by physical location on the slide.



# Create a Flights from all Groups

- All Destinations are associated with a RollCall
- Any associated RollCall and Seats Available should be linked previously by physical location relationship on the slide.

<b>Roll Call</b>	<b>Destination</b>
1925	Destination1
1925	Destination2
1925	Destination3
1925	Destination4
1945	Destination5
1945	Destination6
1945	Destination7

## Finishing up Flight data

- Origin is assumed to be the Space-A terminal that the photo came from.
  - Norfolk originating flights would be on the Norfolk Space-A Facebook page
  - BWI originating flights would be on the BWI Space-A Facebook page

# Fly Space-A Limitations

- The procedural nature of the program sets some limitations on Fly Space-A's capabilities.
  - Roll Call or Seat Available listings may be missed due to OCR and fuzzy matching limitations.
  - Destinations WILL be missed if they are not known beforehand and added to the premade terminal list.
- There is no machine learning implemented in Fly Space-A. However many numerical constants related to bounding boxes, Levenshtein Distance, and terminal names may be tuned through machine learning.



# Future Ideas

- Routing finding can be performed between multiple terminals so that users can compute a multi-leg flight.
  - This does not require using Fly Space-A at all if the user has their own source flight data.

# Acknowledgements

gokogiri by moovweb and jbowtie

latlng by bradfitz

pq - Golang PostgreSQL driver

Fuzzy by Sajari

goprocinfo by c9s

