Lecture 10

Derandomization, quick sort, and stable matching I

Chinmay Nirkhe | CSE 421 Winter 2026

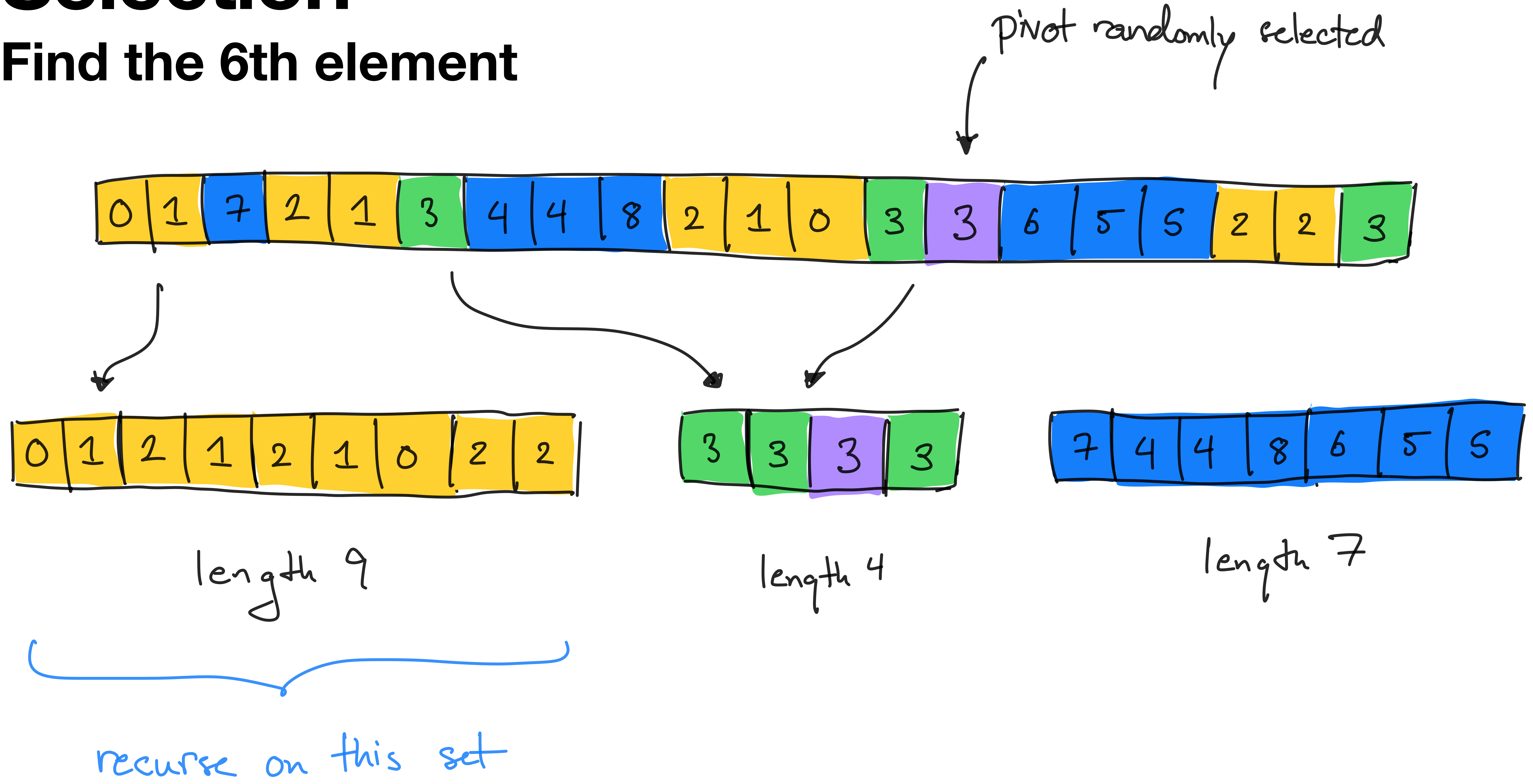
Writing quality

- I’ve been pretty pleased with the solutions for the first two sets of problems
- You may have gotten a -0.01 for writing quality
    - Take the time to chat with TAs about how to improve writing
    - Take a look at the solutions to see how we might write the solutions
    - Emulate solutions from section worksheets
- Starting with set 4, we will increase the deduction for writing quality if egregious

Previously in CSE 421…

Selection

Find the 6th element



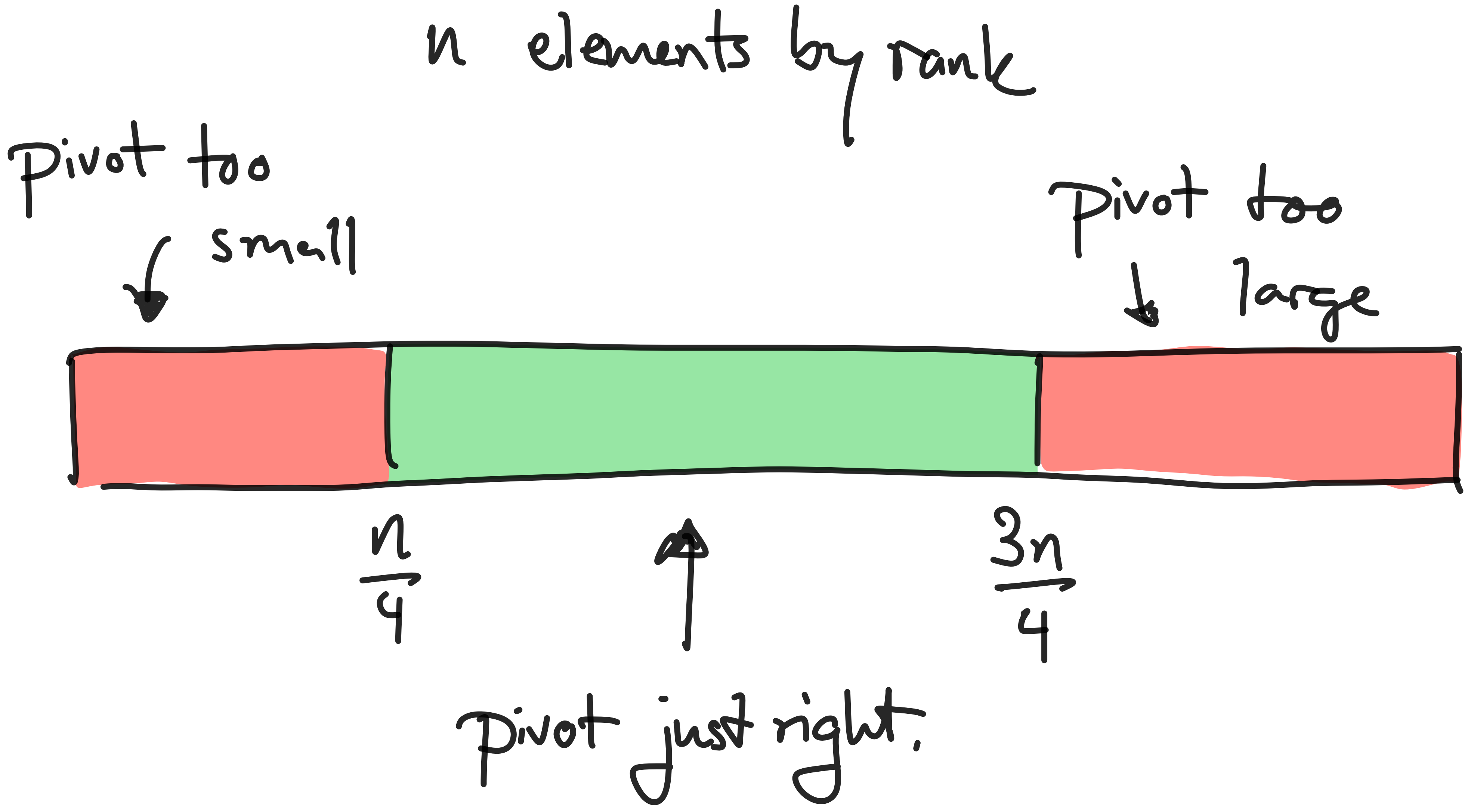
Selection

- **Recursive algorithm** $\text{Selection}(X, k)$**:**
    - Randomly sample j from $[n]$. Call x_j the "**pivot**".
    - Filter X into X_L, X_E, and X_R based on if $x_i < x_j$, $x_i = x_j$, or $x_i > x_j$.
    - If $|X_L| \geq k$, recursively output $\text{Selection}(X_L, k)$.
    - Else if, $|X_L| + |X_E| \geq k$, output x_j.
    - Else, recursively output $\text{Selection}(X_R, k - |X_L| - |X_E|)$.

Runtime analysis

- In order to apply the master theorem, we would need to argue that each recursive call was reducing the input size from n to n/b for $b > 1$
  - $T(n) = T(n/b) + cn \implies T(n) = \frac{c}{1 - 1/b} n$
- However, each call may not reduce the size from n to n/b
- Depends on how close the randomly chosen x_j is to the middle
  - If pivot x_j was the largest element, then $|X_L| = n - 1, |X_E| = 1$, and $|X_R| = 0$.
  - Decreases instance size from n to $n - 1$.
  - Fortunately, the probability this occurs is $1/n$.

Runtime analysis



- **Amortized analysis**:
  - If pivot x_j is the ℓ-th element, then the next problem is of size $\leq \max\{\ell, n-\ell\}$.
  - With probability $\geq 1/2$, pivot x_j is the ℓ-th element for $\ell \in \{n/4,\ldots,3n/4\}$.
  - The expected compute in reducing from n-sized instance to a $3n/4$-sized instance is $O(n)$.
- Total **expected** runtime: $T(n) = T(3n/4) + O(n) \implies T(n) = O(n)$.

Runtime analysis

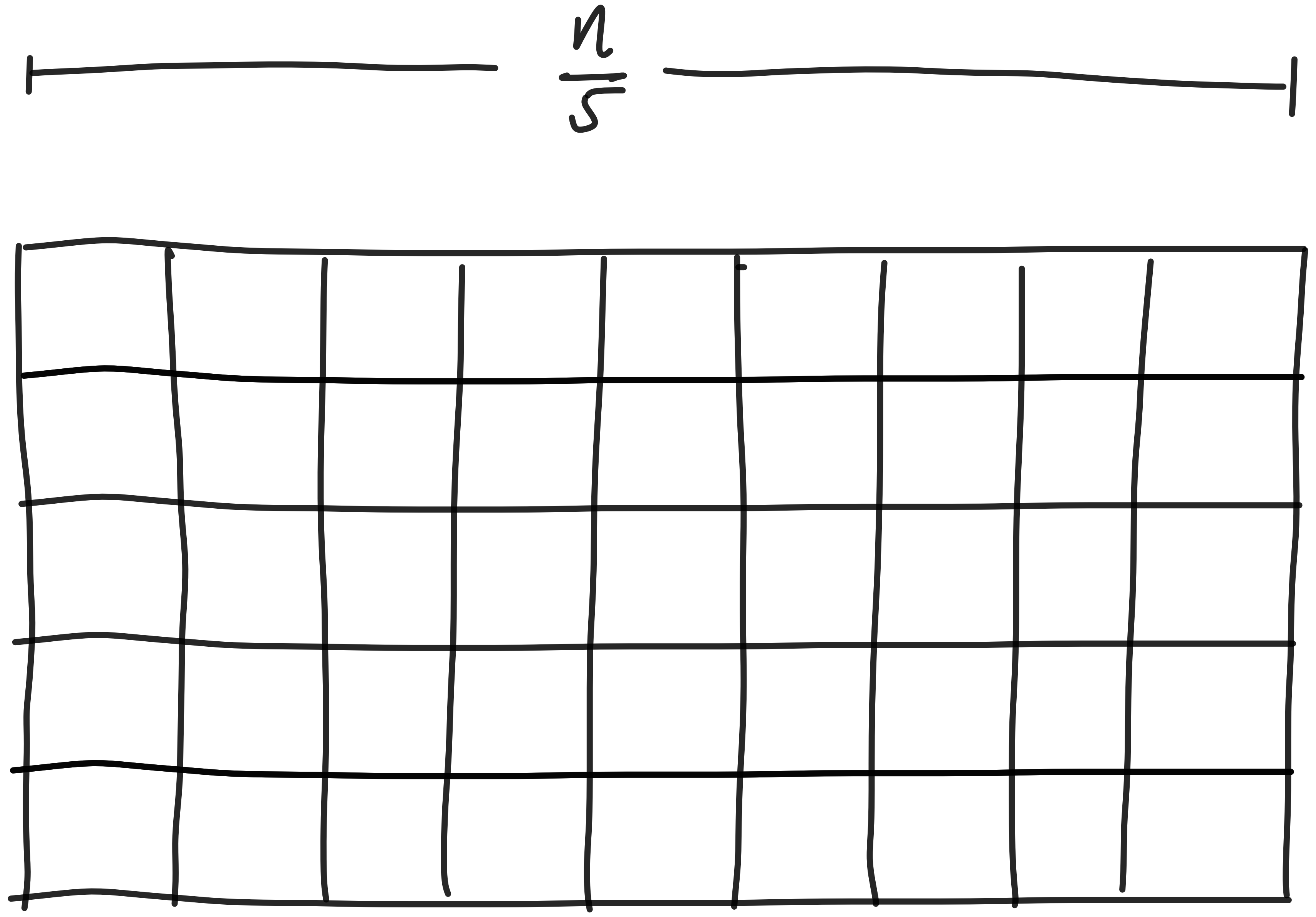
- **Amortized analysis:**
    - If pivot x_j is the ℓ-th element, then the next problem is of size $\leq \max\{\ell, n - \ell\}$.
    - With probability $\geq 1/2$, pivot x_j is the ℓ-th element for $\ell \in \{n/4,\ldots,3n/4\}$.
    - The expected compute in reducing from n-sized instance to a $3n/4$-sized instance is $O(n)$.
        - $\geq 1/2$ probability, shrinks in 1 reduction.
        - $\geq 1/4$ probability, shrinks in 2 reductions.
        - … $\geq 1/2^j$ probability, shrinks in j reductions …
        - Expected compute is $\leq O(n) \cdot \left(\frac{1}{2} + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \ldots\right) = O(n) \cdot 2$
- Total **expected** runtime: $T(n) = T(3n/4) + O(n) \implies T(n) = O(n)$.

Derandomization

- The worst case runtime is $O(n^2)$.
    - Only happens with $2^{-\Omega(n \log n)}$ probability.
- But, is there an algorithm that didn't require randomness?
    - Why?
- Recall, if we could guarantee that the pivot x_j was in the middle half, then each recursion would decrease in size by $3/4$.
- **Blum-Pratt-Floyd-Rivest-Tarjan (1973)**: Calculate a pivot in the middle $4n/10$ in time $O(n)$.

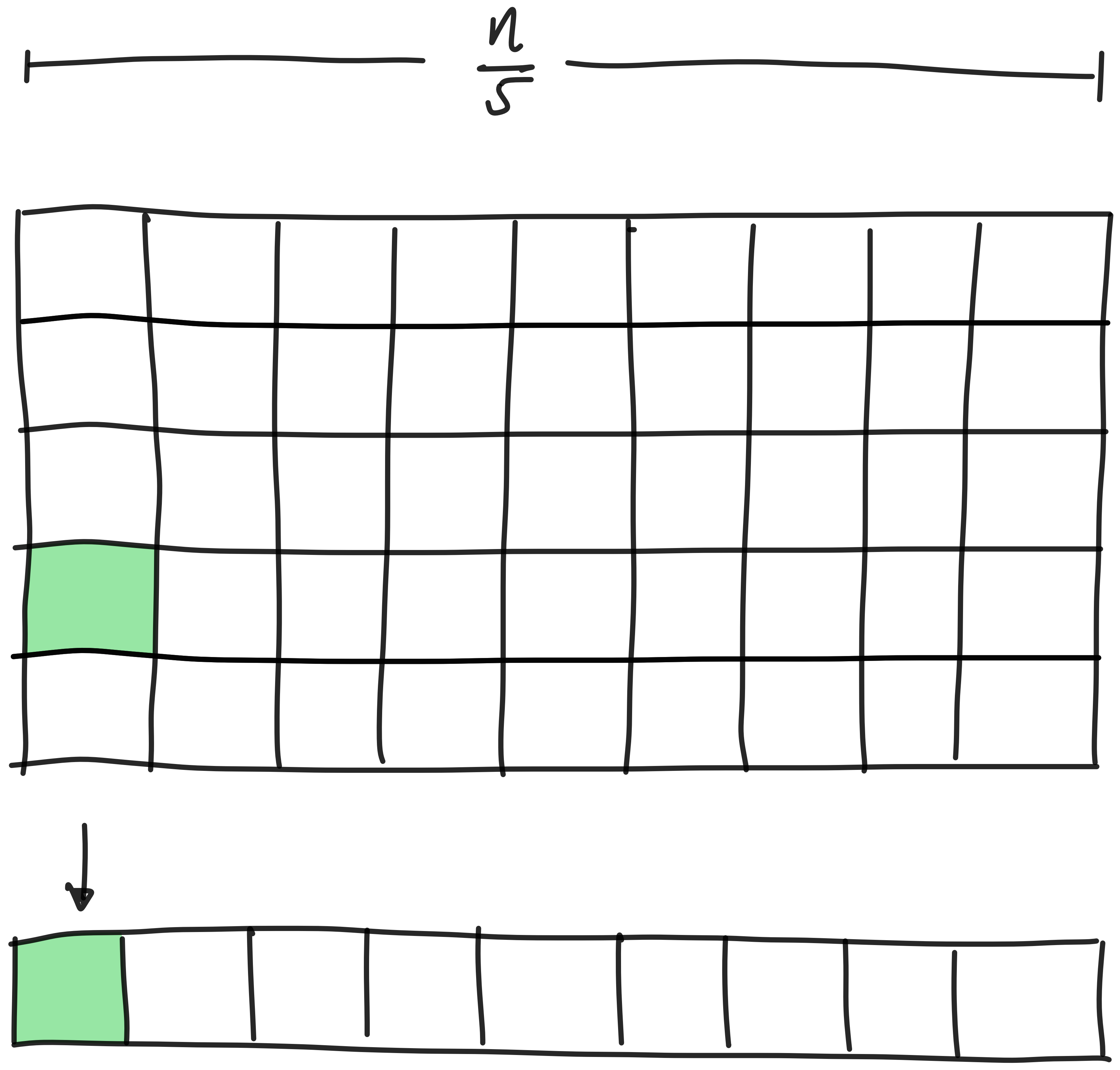
Pivot selection algorithm

- Express the n elements as a $5 \times (n/5)$ matrix of elements



Pivot selection algorithm

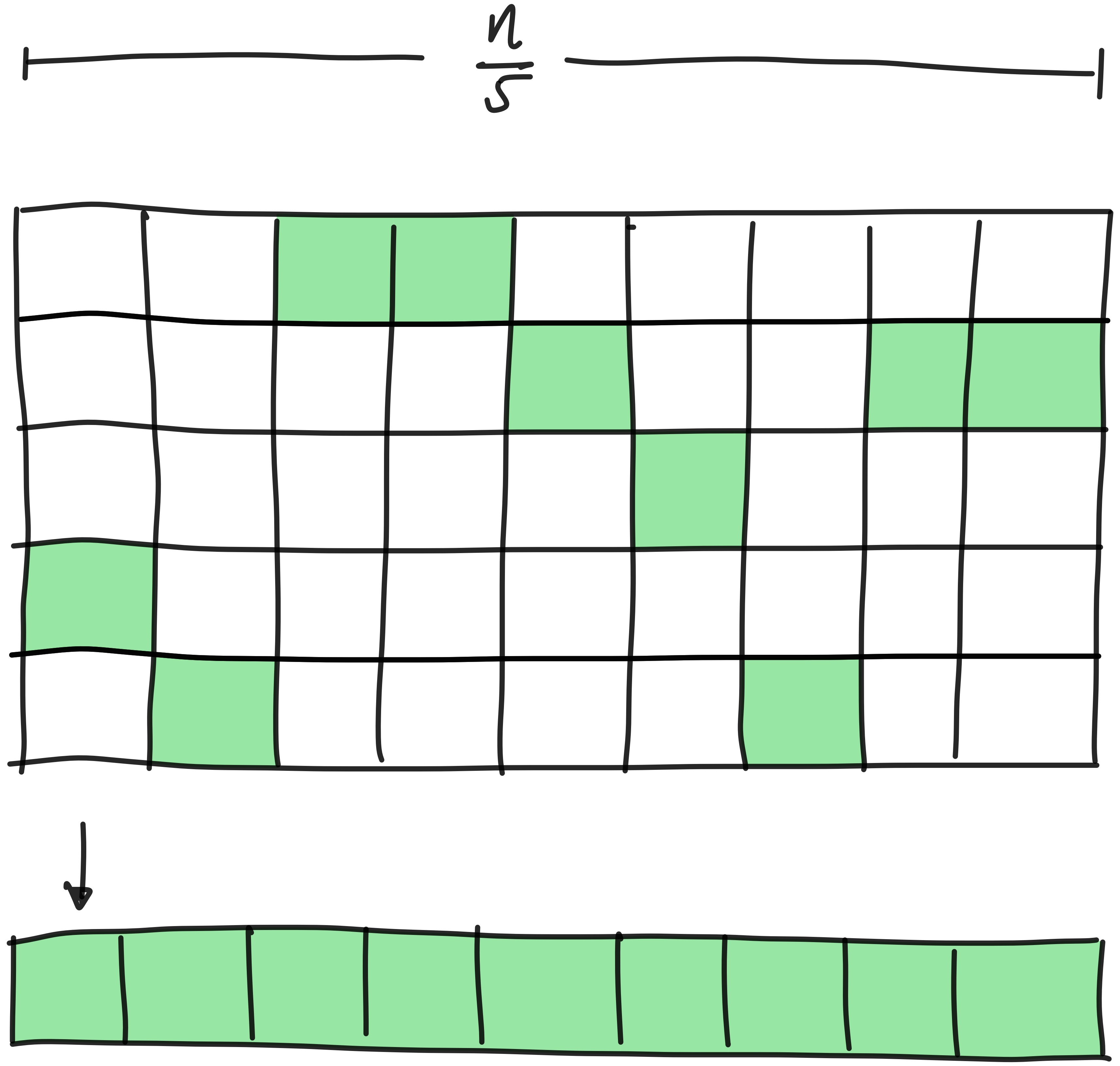
- Express the n elements as a $5 \times (n/5)$ matrix of elements
- Calculate the medians of each of the columns: $Y = (y_1, y_2, \ldots, y_{n/5})$



the median is one of the 5 elements in the column

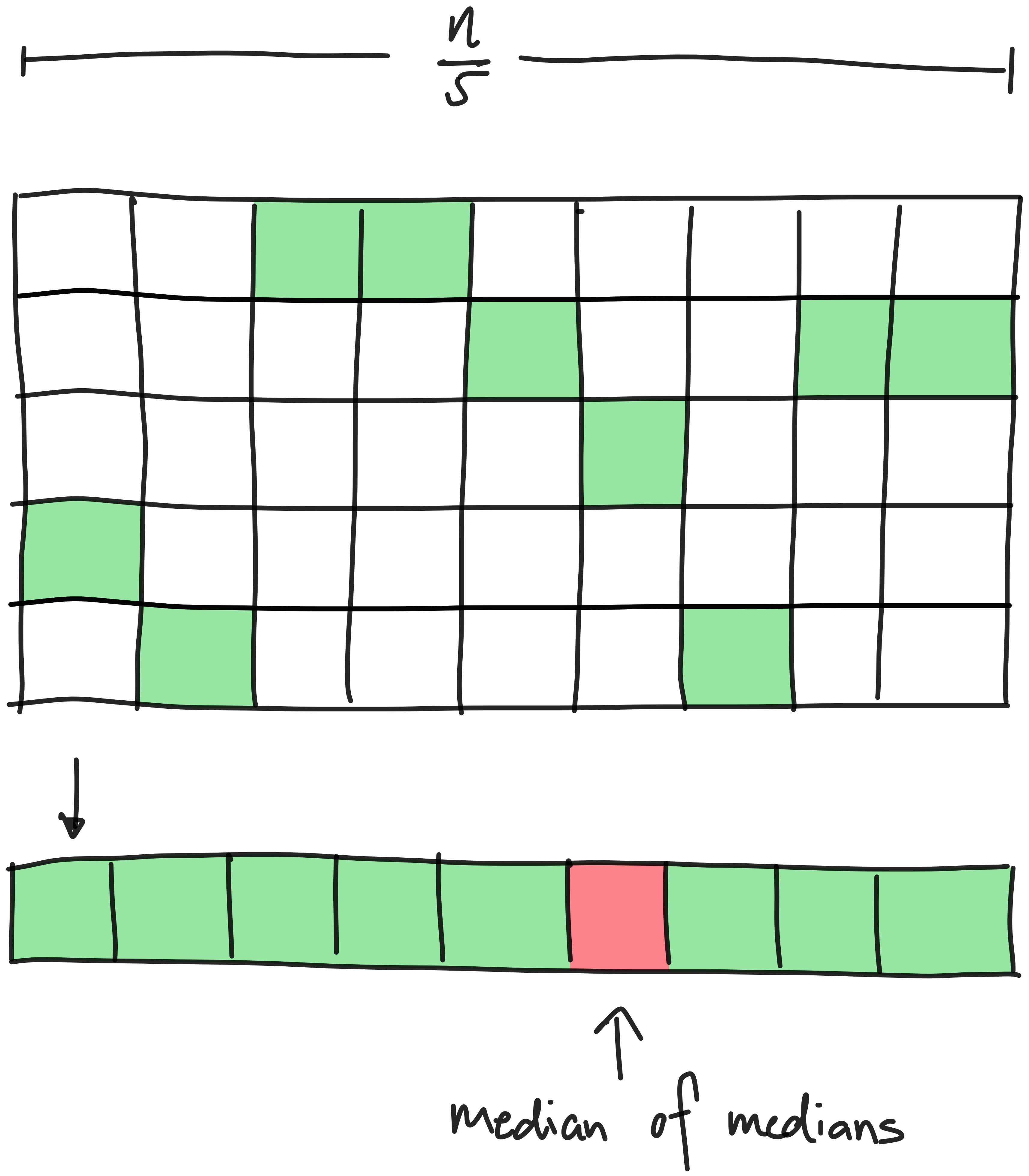
Pivot selection algorithm

- Express the n elements as a $5 \times (n/5)$ matrix of elements
- Calculate the medians of each of the columns: $Y = (y_1, y_2, \ldots, y_{n/5})$



Pivot selection algorithm

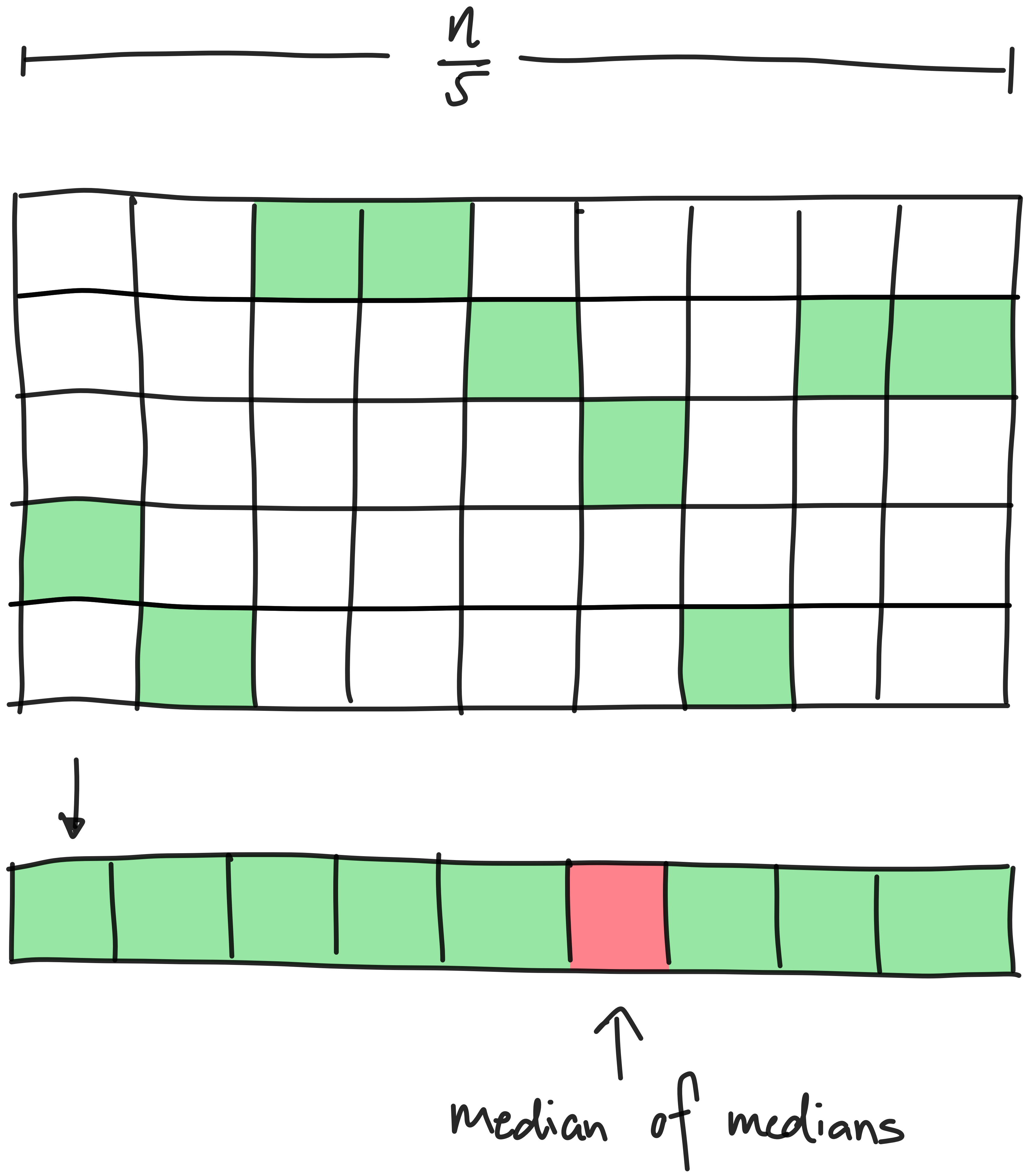
- Express the n elements as a $5 \times (n/5)$ matrix of elements
- Calculate the medians of each of the columns: $Y = (y_1, y_2, \ldots, y_{n/5})$
- Choose the pivot as the median of the medians: $p \leftarrow \text{median}(Y)$



Pivot selection algorithm

Runtime analysis

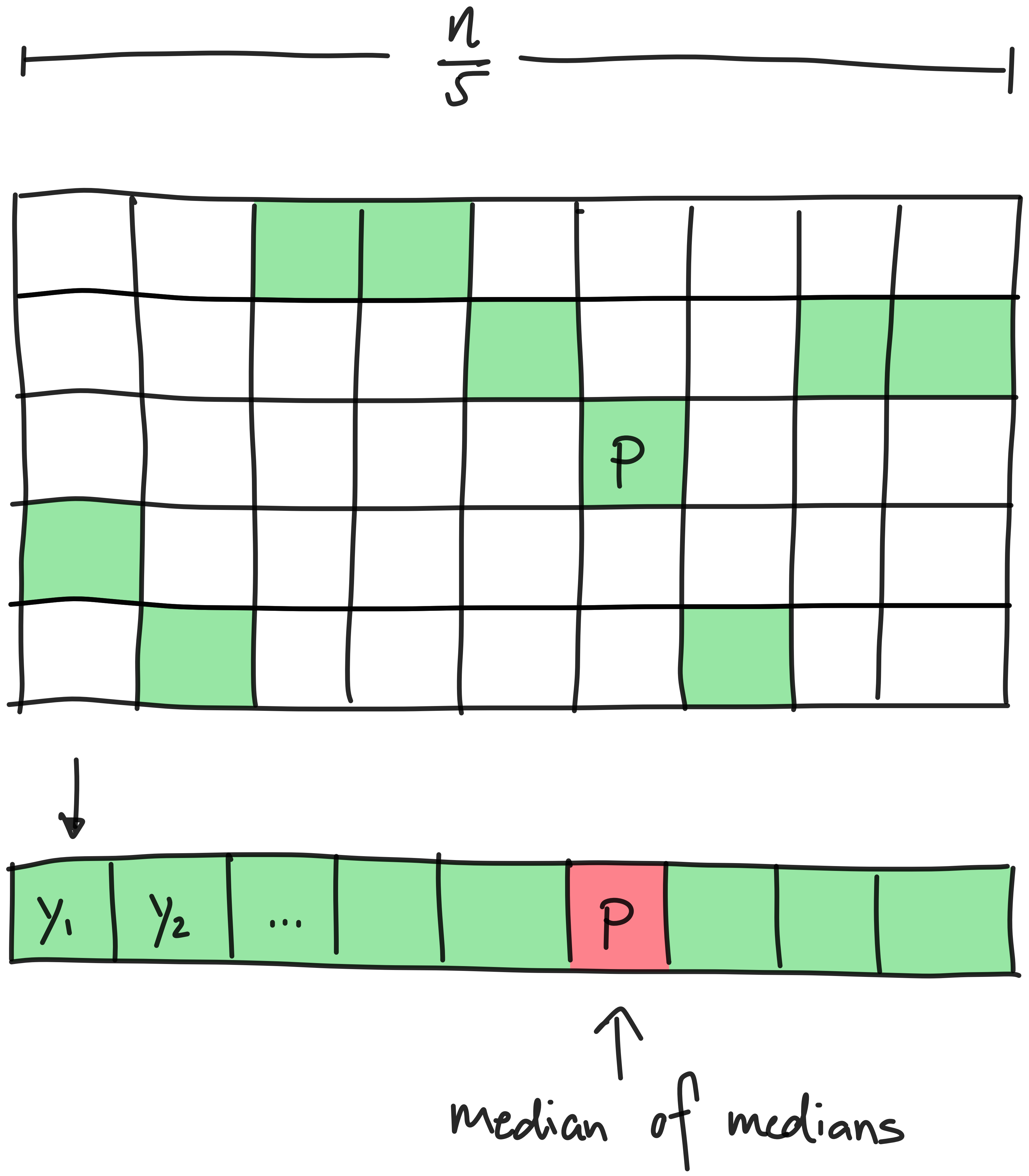
- Express the n elements as a $5 \times (n/5)$ matrix of elements
  Only semantic.
- Calculate the medians of each of the columns: $Y = (y_1, y_2, \ldots, y_{n/5})$
  ← O(1) per col.
  Total O(n).
- Choose the pivot as the median of the medians: $p \leftarrow \text{median}(Y)$ T(n/5) recursively



Total time: $T(n) = T(\frac{n}{5}) + O(n) \Rightarrow T(n) = O(n)$.

Pivot selection algorithm

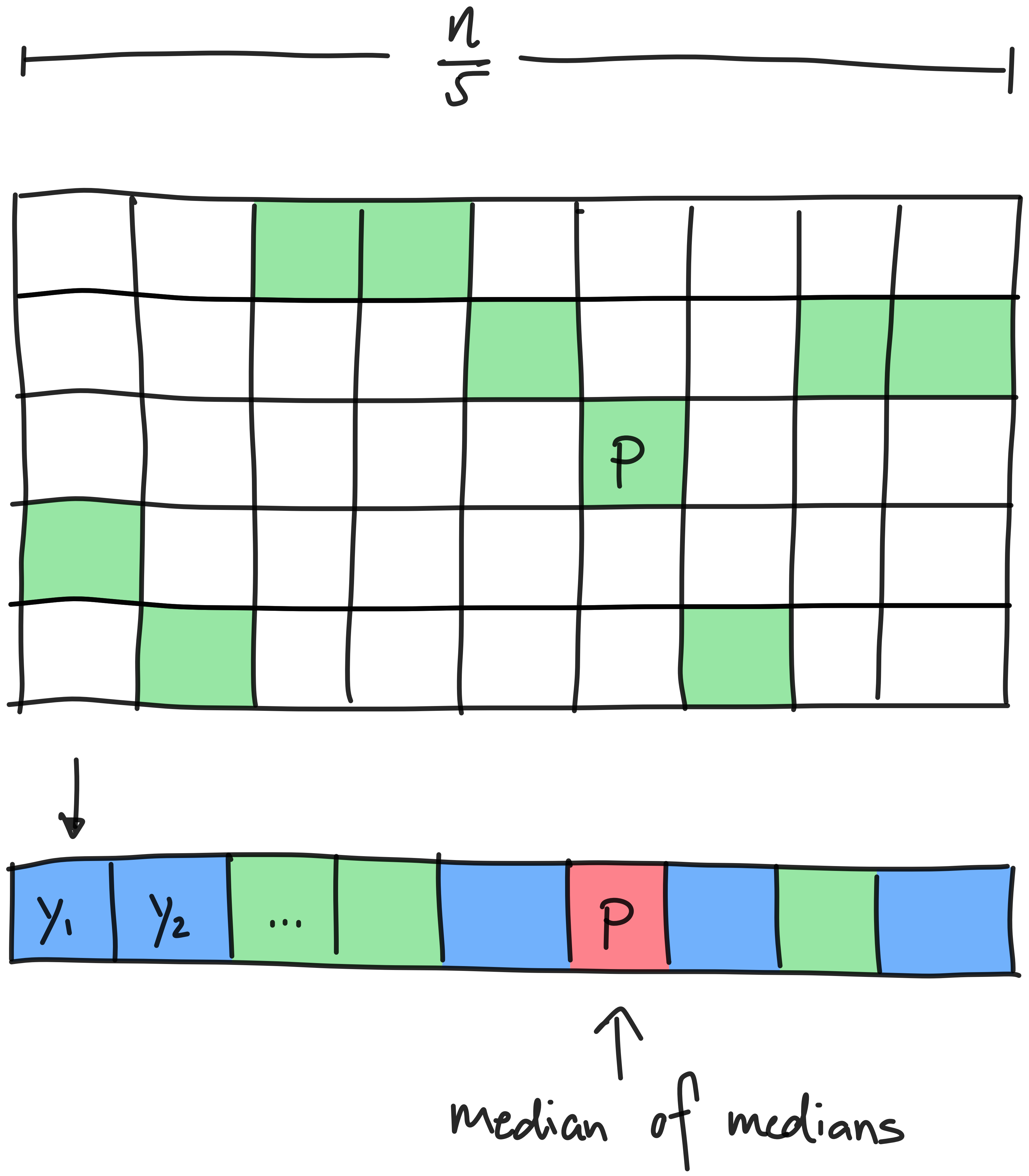
Proof of correctness



Pivot selection algorithm

Proof of correctness

- There are $\geq n/10$ columns such that $y_j \geq p$.

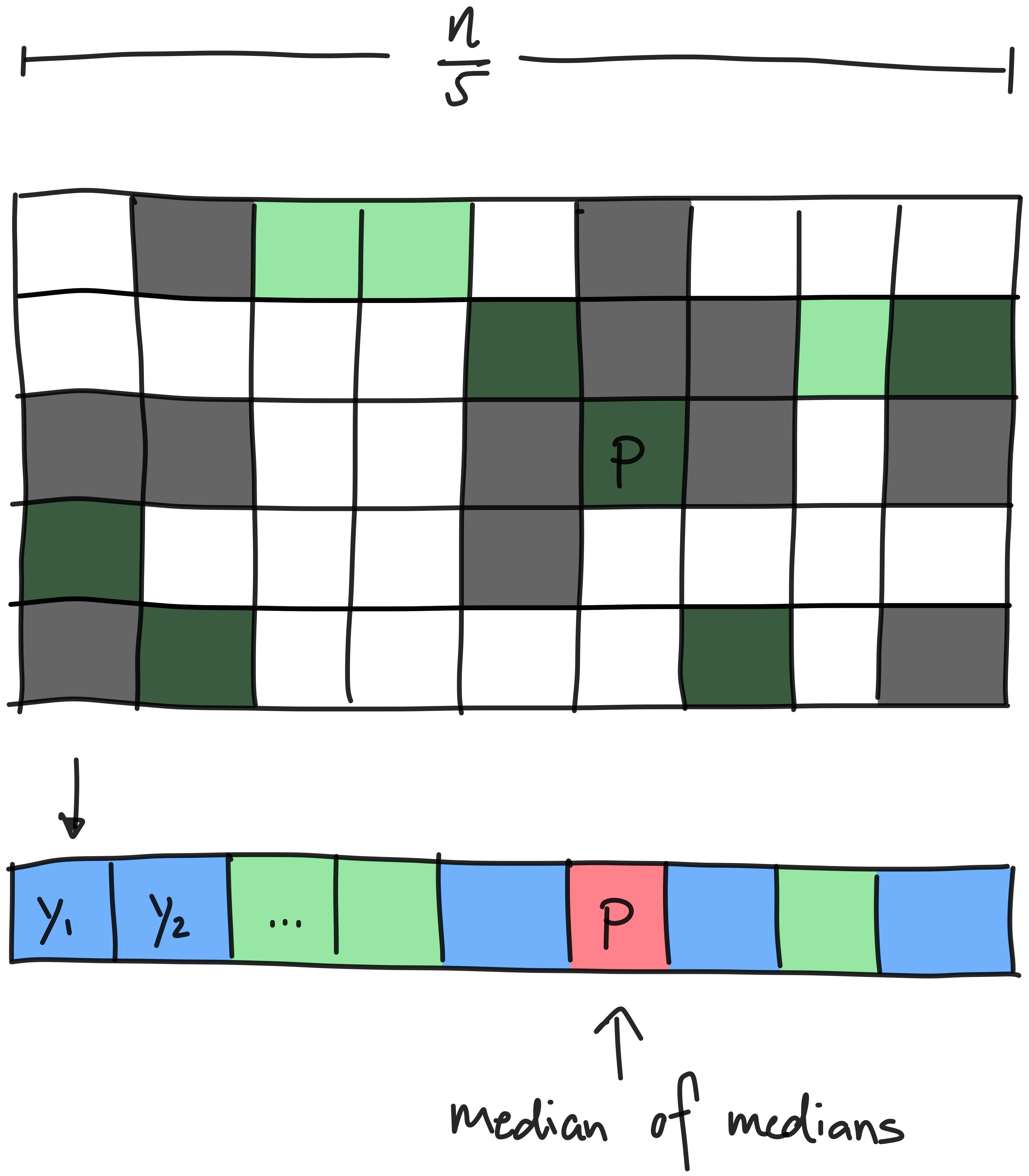


Pivot selection algorithm

Proof of correctness

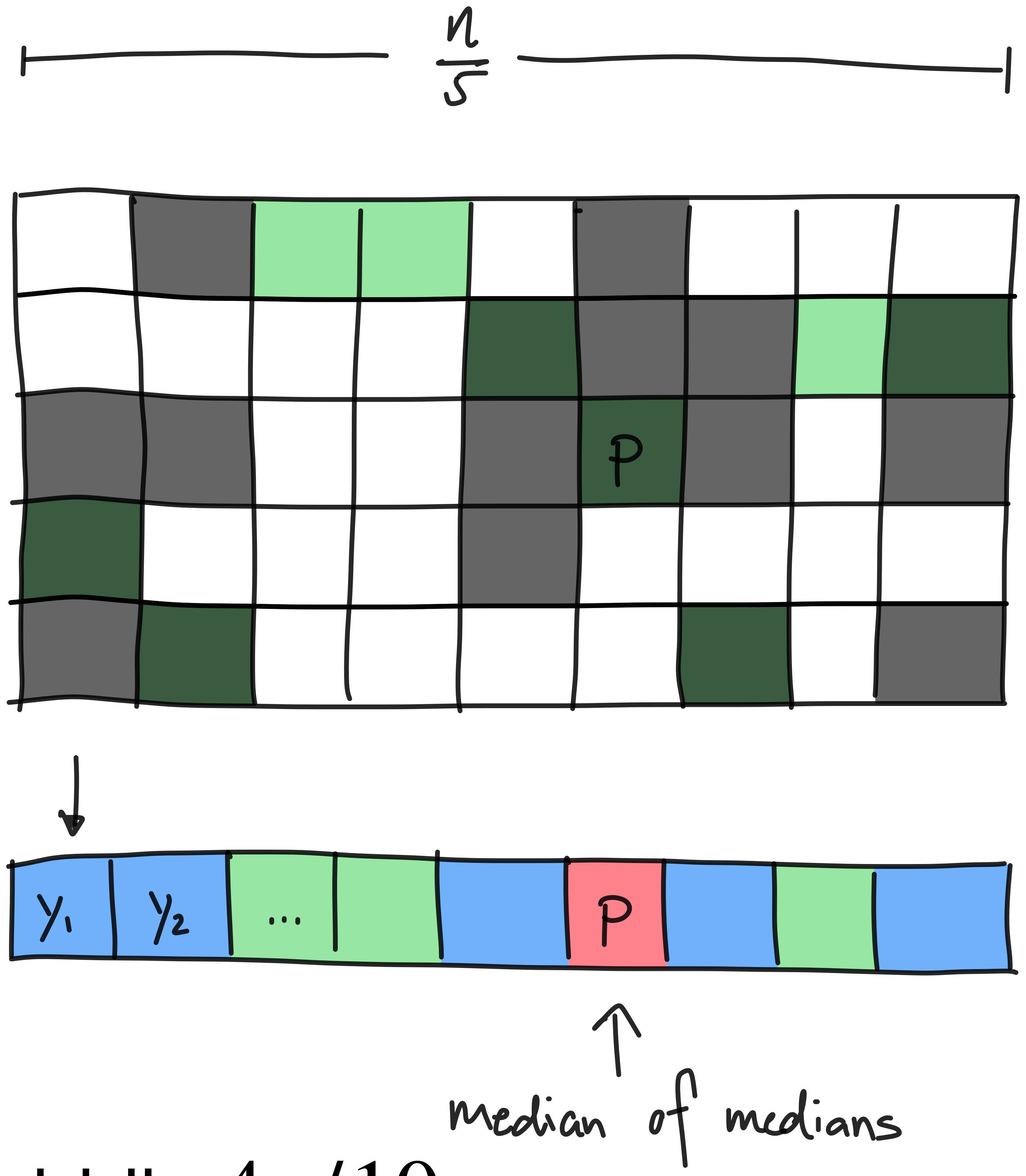
- There are $\geq n/10$ columns such that $y_j \geq p$.

- In each such column, there are 3 elements $\geq y_j$.



Pivot selection algorithm

Proof of correctness



- There are $\geq n/10$ columns such that $y_j \geq p$.

- In each such column, there are 3 elements $\geq y_j$.

- Therefore, there are $\geq 3n/10$ elements $\geq p$.

- Similarly, there are $\geq 3n/10$ elements $\leq p$.

- So, p is in the middle $4n/10$ elements and a good pivot.

Median/Selection algorithm

Total: $T(n) = T\left(\frac{7}{10}n\right) + T\left(\frac{n}{5}\right) + O(n)$

$\Rightarrow T(n) = O(n)$

↑ Pset problem on how to analyze this generalization of Master theorem

- **Input:** $(X, k) \in \mathbb{R}^n \times [n]$
- **Output:** the k-th item in the list X
- **Algorithm:**
  - Calculate $p \leftarrow$ median-of-medians(X) in a $5 \times (n/5)$ division.
  - Filter X into X_L, X_E, and X_R based on p
  - If $|X_L| \geq k$, recurse $\text{Selection}(X_L, k)$
    - Else if $|X_L| + |X_E| \geq k$, return p
    - Else, return $\text{Selection}(X_R, k - |X_L| - |X_E|)$.

← recursive $T\left(\frac{n}{5}\right) + O(n)$

recursive $T\left(\frac{7}{10}n\right)$

Quicksort algorithm

- The algorithm we just analyzed, “Quickselect”, can be generalized to sorting
- **Sorting algorithm** $\mathrm{Quicksort}(X)$:
    - Pick a pivot p (either randomized or with median-of-medians)
    - Filter X into X_L, X_E, X_R by comparing elements with p
    - Concatenate $\mathrm{Sort}(X_L), X_E, \mathrm{Sort}(X_R)$.

Computing expected runtime is challenging due to variable size

Quicksort algorithm

Runtime analysis

- Runtime depends on pivot selection
- **Median-of-medians:**
  - $T(n) \leq T(\alpha n) + T(n - \alpha n) + O(n)$ for $\alpha \in [0.3, 0.7]$
  - $T(n) = O(n \log n)$ by analysis in problem 54
- **Choose random element:**
  - Worst case: $O(n^2)$ time
  - Amortized: $O(n \log n)$ (next!)

Quicksort algorithm

Runtime analysis

[for random choice of pivot]

- **Observations**:
  - The runtime of Quicksort is proportional to the number of comparisons
  - The algorithm only compares two elements if one is the pivot
- Let $Y = (y_1, \ldots, y_n)$ be the sorted version of the input.
- Let $p_{ij} = \mathbf{Pr}\left[y_i \text{ and } y_j \text{ are compared}\right]$
- **Claim:** $p_{ij} \leq \dfrac{2}{j-i+1}$ when $i < j$.

Expected number of comparisons:

$$\sum_{i<j} P_{ij} \leq 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{j-i+1}$$

$$= 2 \sum_{i=1}^{n} \sum_{k=1}^{n-i+1} \frac{1}{k+1}$$

$$= 2 \sum_{i=1}^{n} \underbrace{\left(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-i+1}\right)}_{\substack{\leq \log(n-i+1) + 1 \\ \leq \log(n) + 1}}$$

$$\leq 2 n \log n + 2n.$$

Runtime of quicksort $= O(n \log n)$.

Proof of claim

- **Claim:** $p_{ij} \leq \dfrac{2}{j-i+1}$ when $i < j$.
- **Proof:**
  - $y_i \leq y_j$ and y_i and y_j are compared at most once
    - Comparisons only occur when one of them is the pivot
    - Case 1: $y_i, y_j \in X_E$ and we never recurse on X_E
    - Case 2: $y_i \in X_E, y_j \in X_R$ and we never compare between X_L, X_E, and X_R
    - Case 3: $y_i \in X_L, y_j \in X_E$ and we never compare between X_L, X_E, and X_R

- If and when y_i and y_j are compared during $\mathrm{sort}(X')$ then $y_i, y_{i+1}, y_{i+2}, \ldots, y_j \in X'$
  - Can be formally proven via induction
  - So $|X'| \geq j - i + 1$.
  - Probability that either y_i or y_j is chosen as pivot is $\leq \dfrac{2}{j-i+1}$.

Sorting in the real world

- **Quicksort**
  - Fast almost always, especially for in-memory sorting.
  - Works well with caches due to good locality of reference.
  - In practice,
    - Don't filter X_L, X_E, and X_R. Use in-place swaps.
    - When n is small, insertion sorting is a better base case.
    - Pick pivot randomly for small n, median of 3 random values for medium n, and median-of-medians on 9 elements for large n
    - Never actually run the median-of-medians pivot finding routine

Sorting in the real world

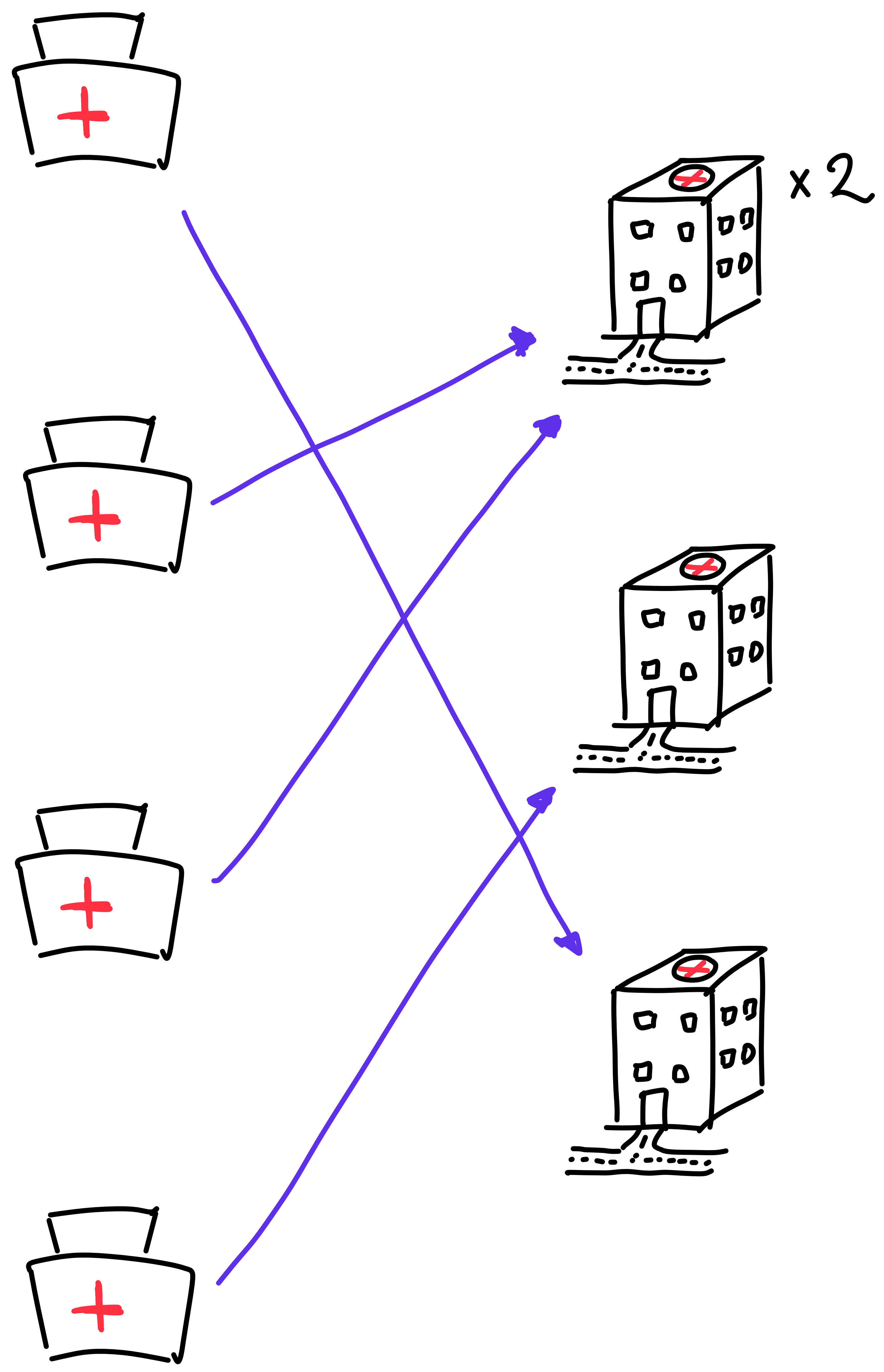
- **Mergesort**
    - Used when data is expressed as a linked list and RAM access to entries in the middle of the list is non-existent
    - Sorting over a dataset that cannot be stored in memory
    - Uses $O(n)$ extra space when sorting arrays over Quicksort

Sorting in the real world

- **Insertion sort**
    - Best when data is almost sorted already
    - $O(n^2)$ when far from sorted
- **Heap sort** - memory efficient choice
- **Bucket sort** - distribution aware sorting
- Etc…

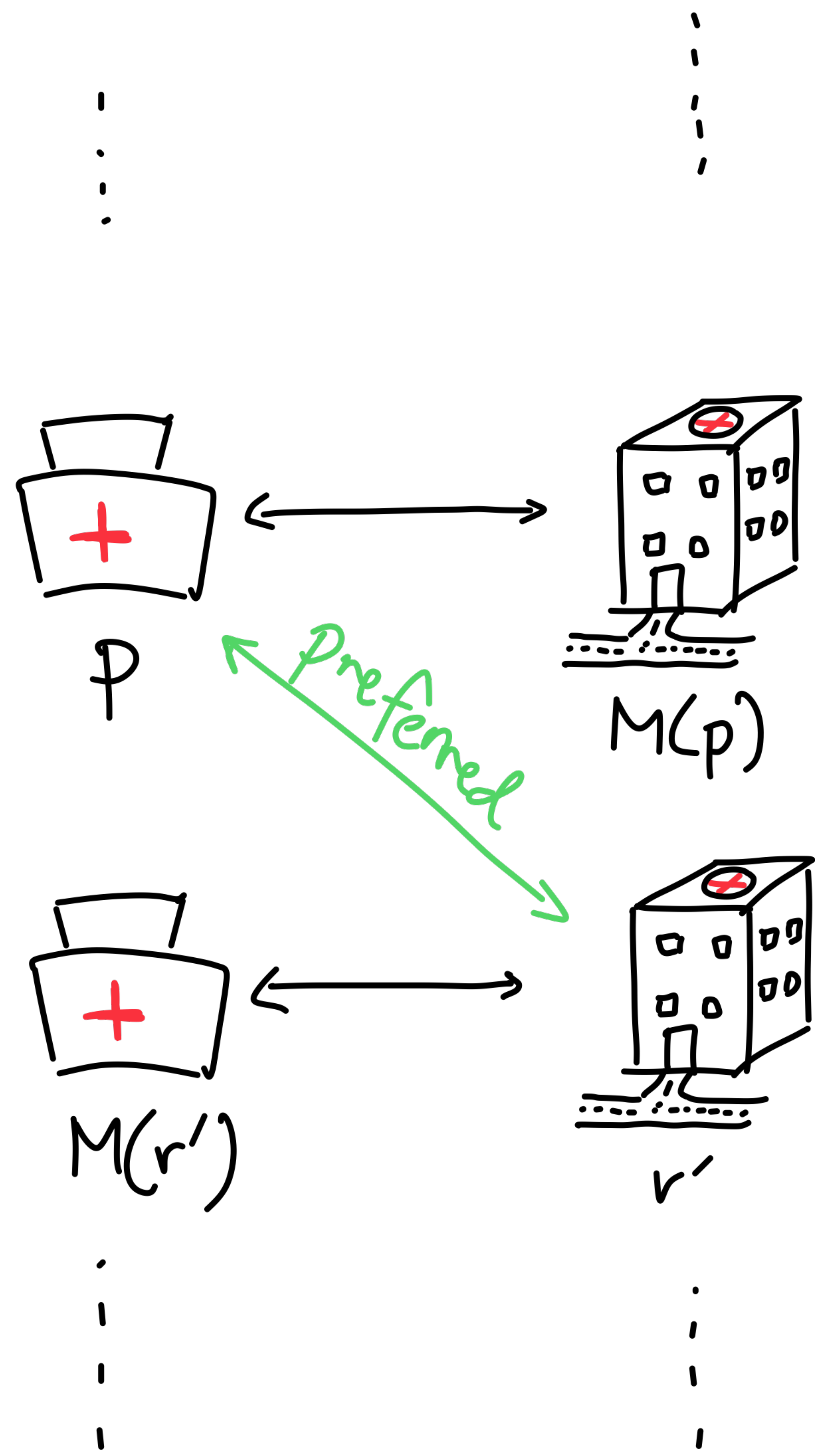
The matching problem

- **Goal**: Given a set of preferences amongst hospital and residents, design an admissions process to allocate residents to hospitals.
- What might we want to optimize for?
- When do we know we have achieved the optimal solution?
- What properties does our optimal solution have?



A notion of *stability*

- Lets assume there are n residents and n hospitals for now.
- A matching M is n disjoint pairs (p, r) assigning hospital r to resident p.
- A resident-hospital pair (resident p, hospital r') is **unstable** for M if both
  - resident p prefers hospital r' to their assigned hospital $M(p)$.
  - hospital r' prefers resident p to their assigned resident $M(r')$.
- A matching is **stable** if the matching has no **unstable** pairs.
  - Natural and desirable condition. *Self-interest* will prevent side-deals from being made.



Cartoon of unstable (p,r')

Can we design an algorithm to find a stable matching?

And does a stable matching necessarily exist?

- **Input to the problem**:
  - Two groups of n people: one group P and the other group R.
  - For each $p \in P$, a ranking from 1 to n of the group R.
  - For each $r \in R$, a ranking from 1 to n of the group P.
- **Output of the problem**:
  - A list of n disjoint pairs M. The matching should be stable with respect to the input rankings.

favorite ↓

least fav. ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Example 1: Is the following matching stable?

X ↔ C

Y ↔ B

Z ↔ A

favorite

least fav.

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Example 1: Is the following matching stable?

No.

X ⟷ C

Y ⟷ B

Z ⟷ A

favorite

least fav.

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

mutually preferred change

Example 2: Is the following matching stable?

X ↔ A

Y ↔ B

Z ↔ C

favorite

least fav.

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Example 2: Is the following matching stable?

YES.

X ↔ A
Y ↔ B
Z ↔ C

favorite ↓ least fav. ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

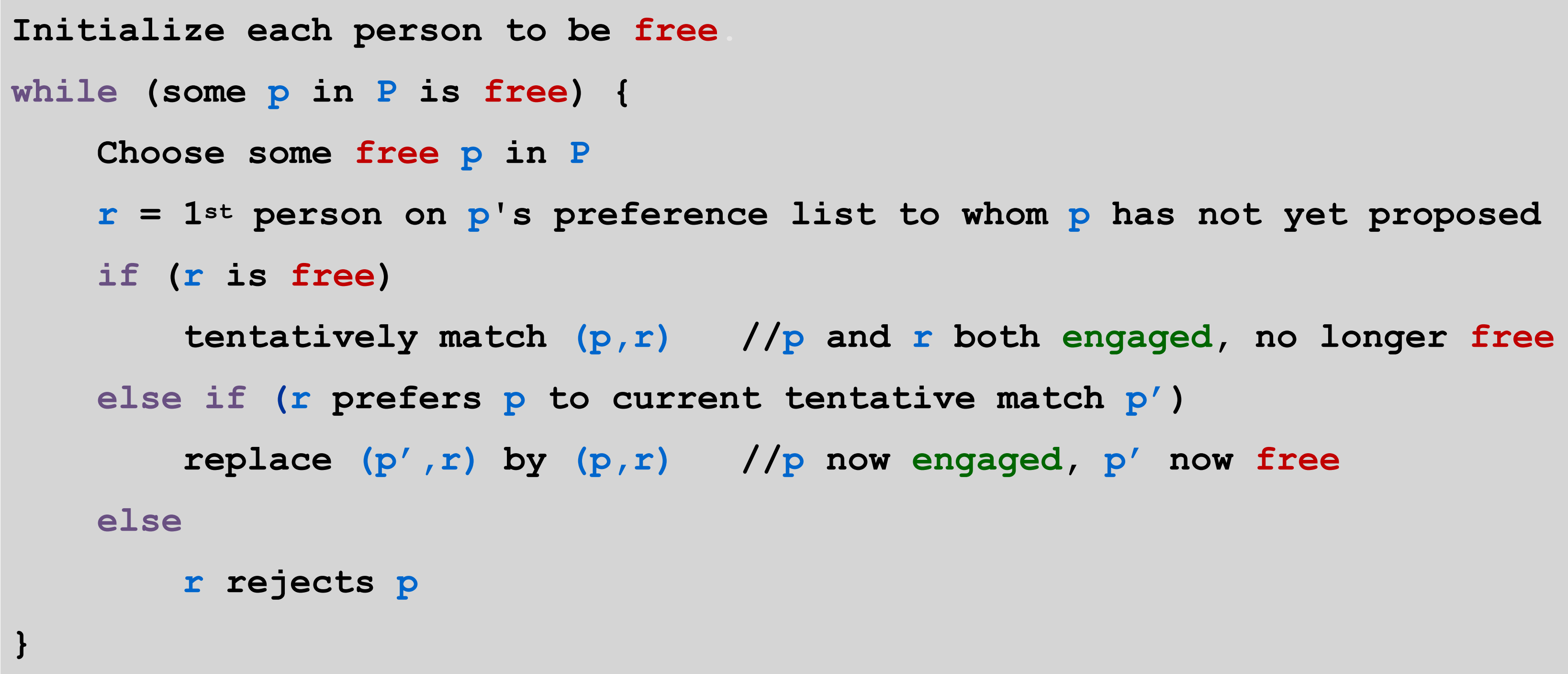
The propose and reject algorithm

Gale & Shapley 1962

The group P proposes and the group R receives

```
Initialize each person to be free.
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p’)
        replace (p’,r) by (p,r)    //p now engaged, p’ now free
    else
        r rejects p
}
```

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

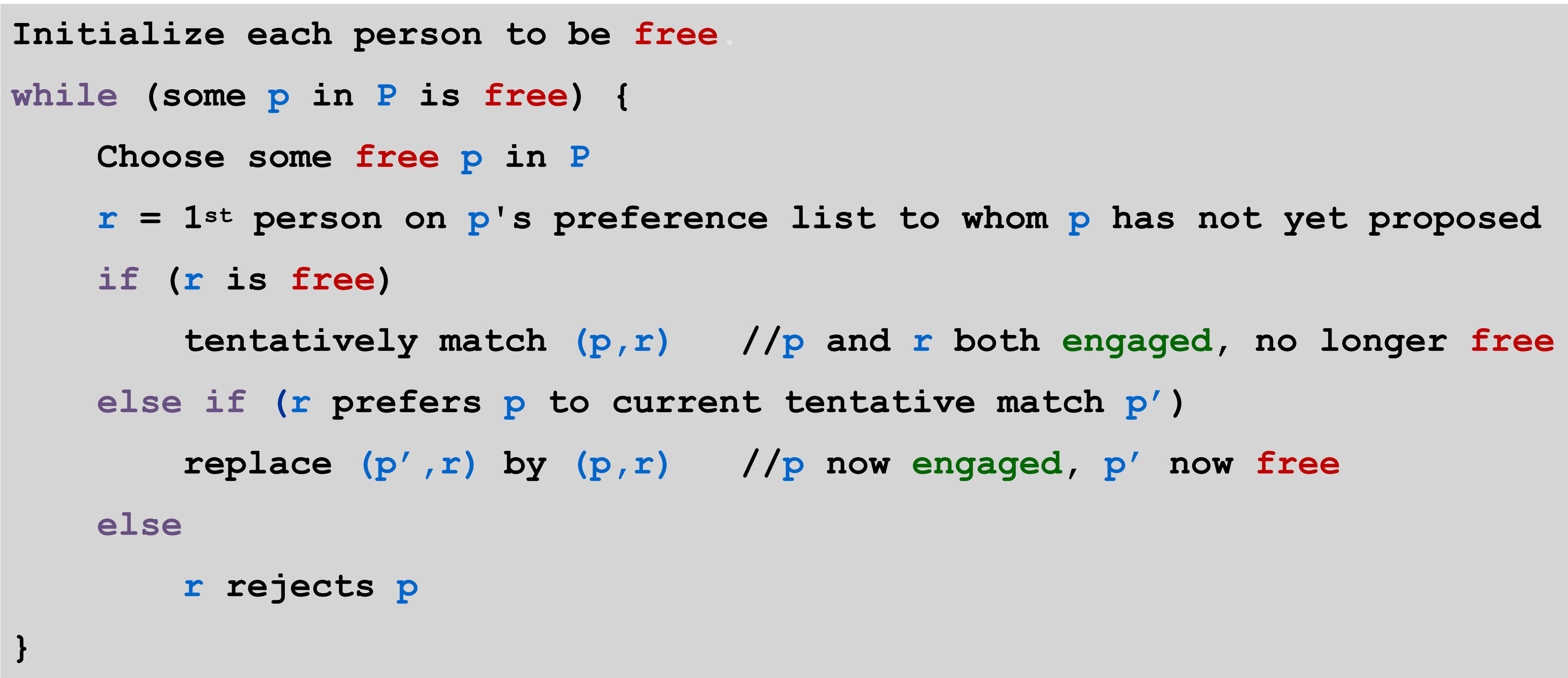
$n = 4$.

We will walkthrough alg, staying blind to the remainder of the input until we have queried it.

	FAV ↓			LEAST ↓
ALPHA				
BRAVO				
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

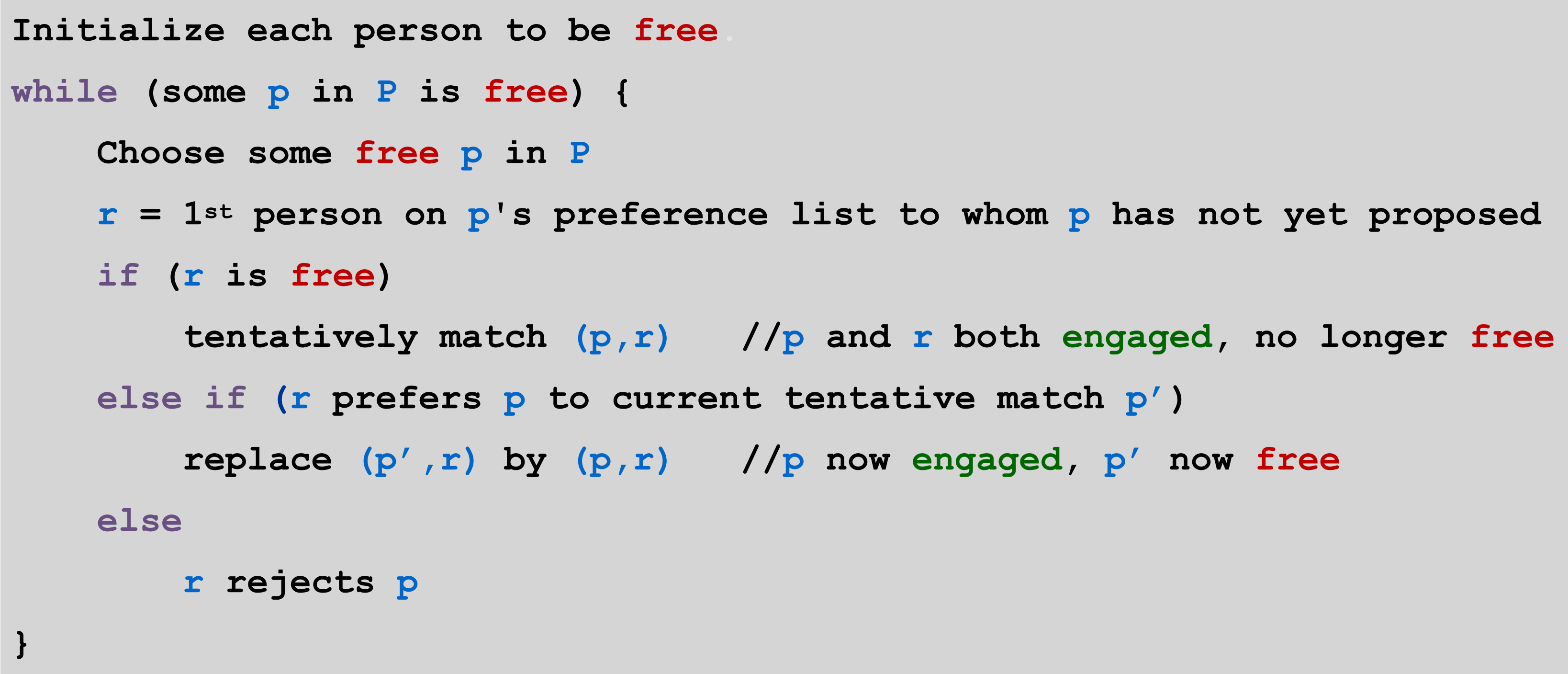
ALPHA	
BRAVO	
CHARLIE	
DELTA	

PAPA	
QUEBEC	
ROMEO	
SIERRA	

	FAV ↓			LEAST ↓
ALPHA				
BRAVO				
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

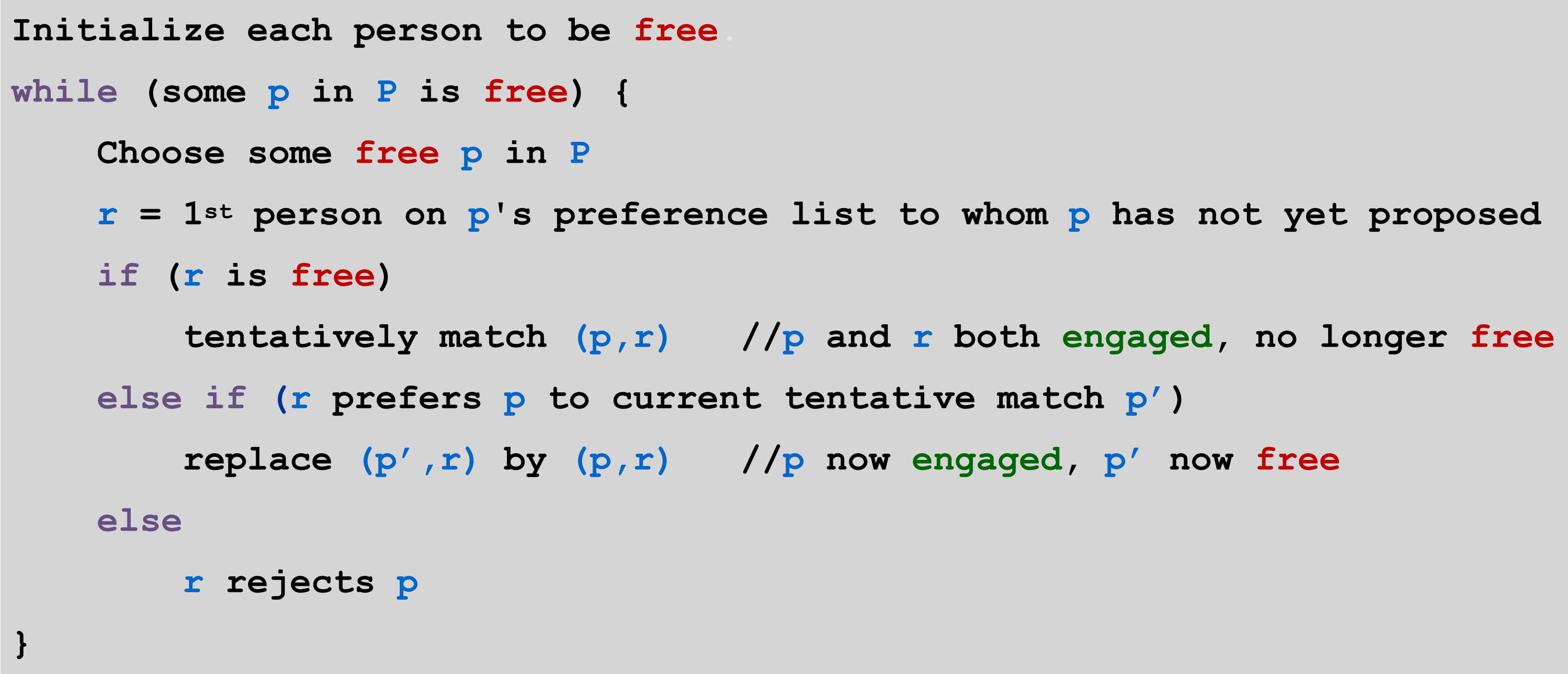
ALPHA	F
BRAVO	F
CHARLIE	F
DELTA	F

PAPA	F
QUEBEC	F
ROMEO	F
SIERRA	F

	FAV ↓			LEAST ↓
ALPHA				
BRAVO				
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

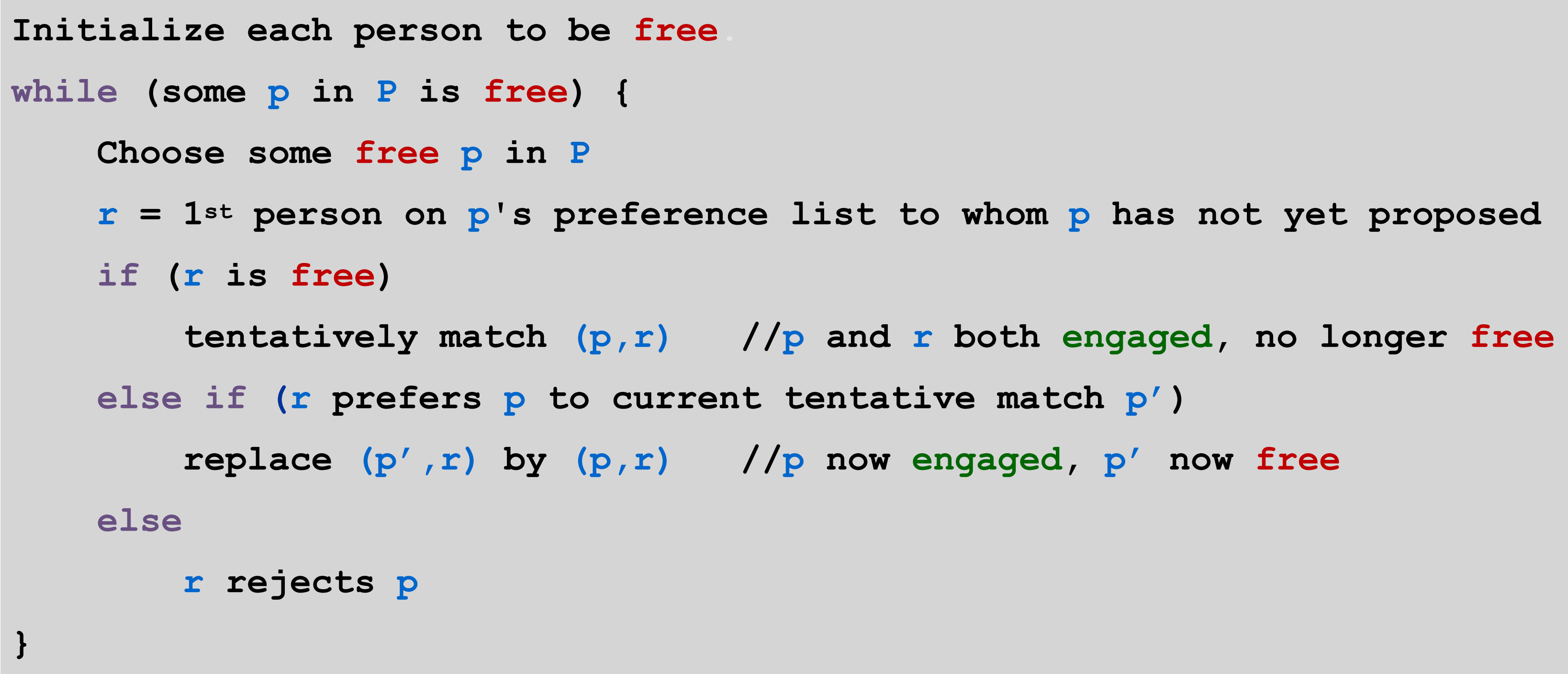
ALPHA	F
BRAVO	F
CHARLIE	F
DELTA	F

PAPA	F
QUEBEC	F
ROMEO	F
SIERRA	F

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO				
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

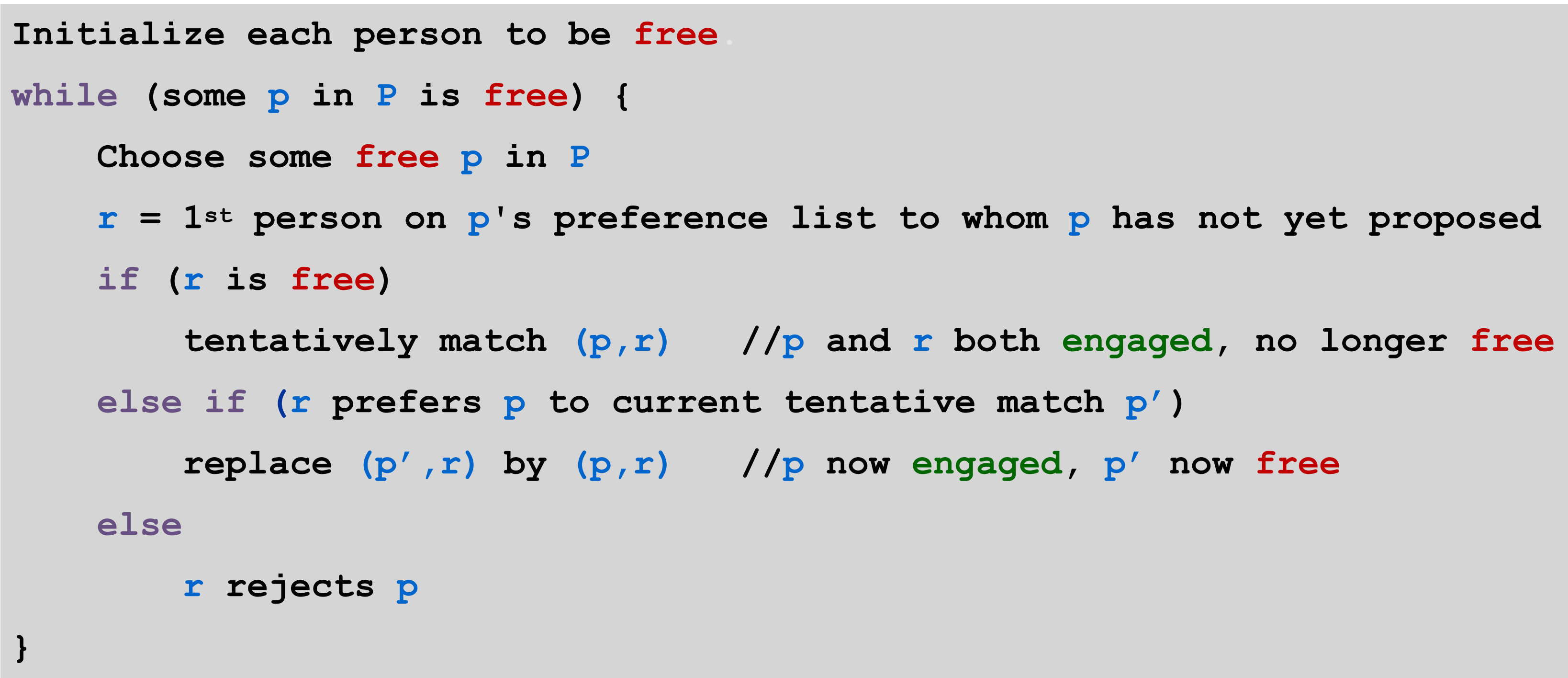
ALPHA	R
BRAVO	F
CHARLIE	F
DELTA	F

PAPA	F
QUEBEC	F
ROMEO	A
SIERRA	F

	FAV			LEAST
ALPHA	R			
BRAVO				
CHARLIE				
DELTA				

	FAV			LEAST
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	F
CHARLIE	F
DELTA	F

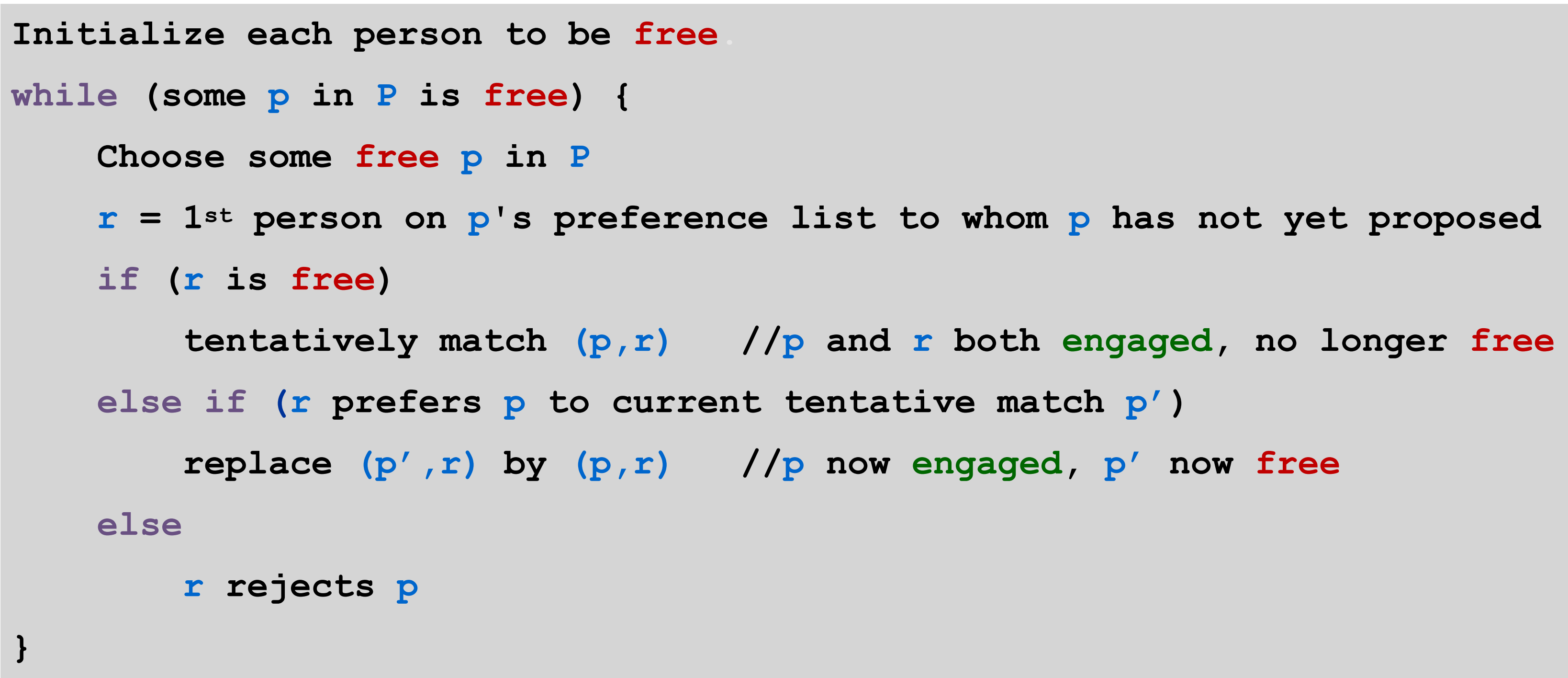
PAPA	F
QUEBEC	F
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO				
CHARLIE				
DELTA				

	FAV			LEAST
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	F
CHARLIE	F
DELTA	F

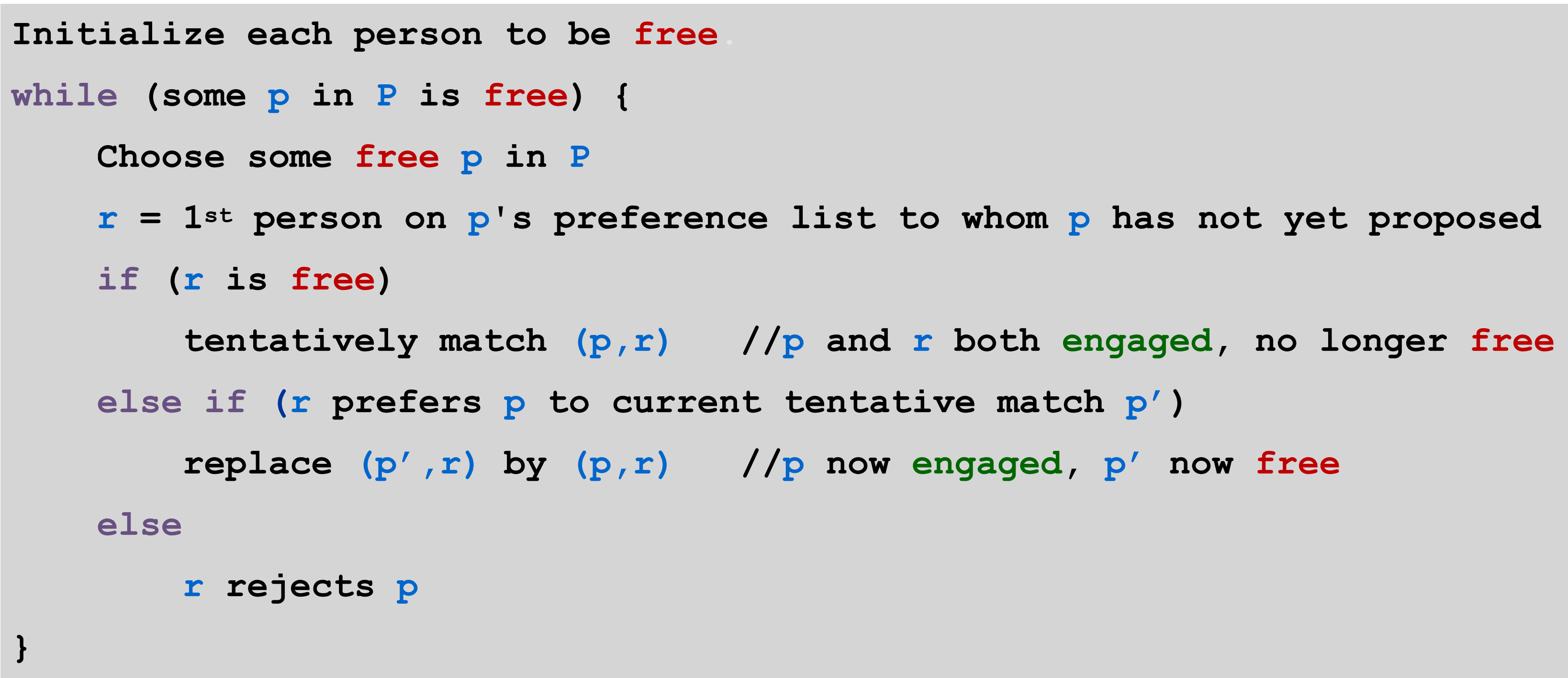
PAPA	F
QUEBEC	F
ROMEO	A
SIERRA	F

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO	Q			
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	Q
CHARLIE	F
DELTA	F

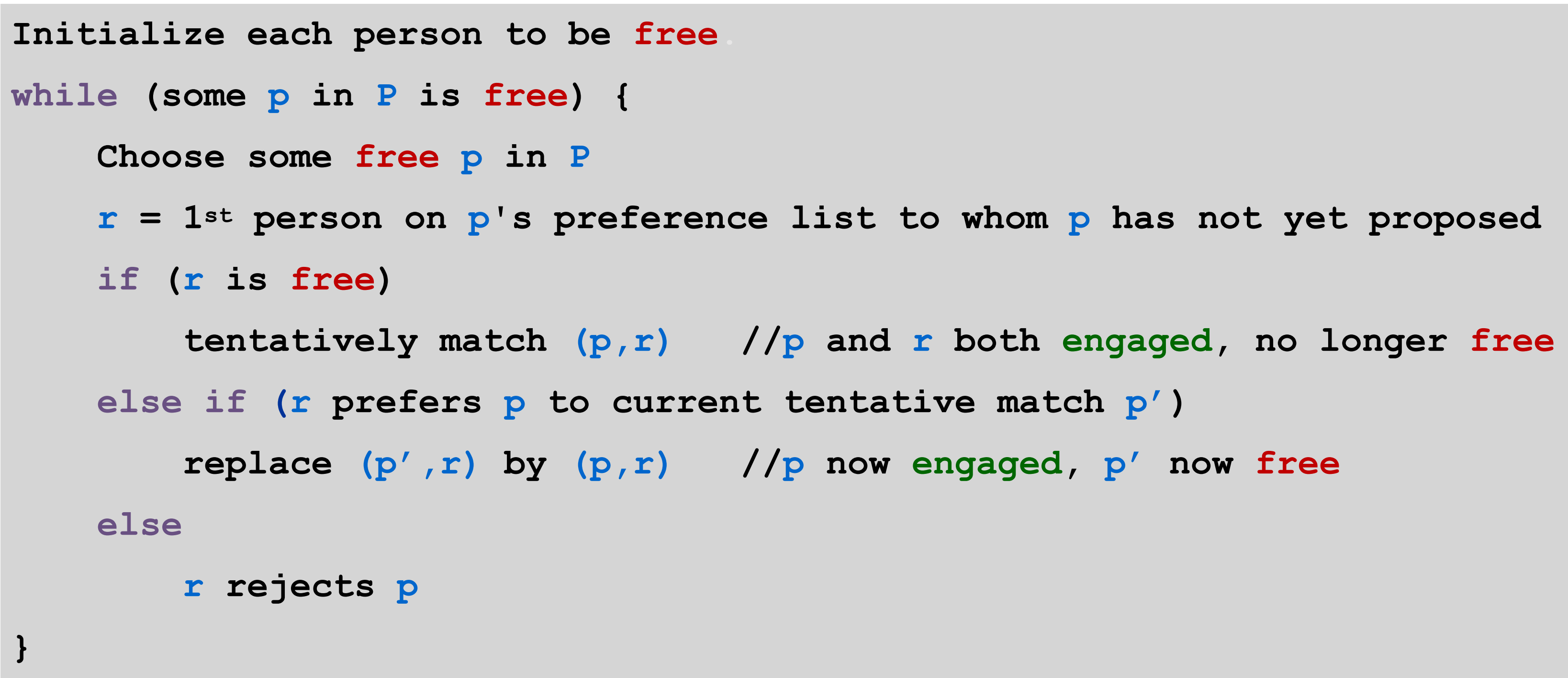
PAPA	F
QUEBEC	B
ROMEO	A
SIERRA	F

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO	Q			
CHARLIE				
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	Q
CHARLIE	F
DELTA	F

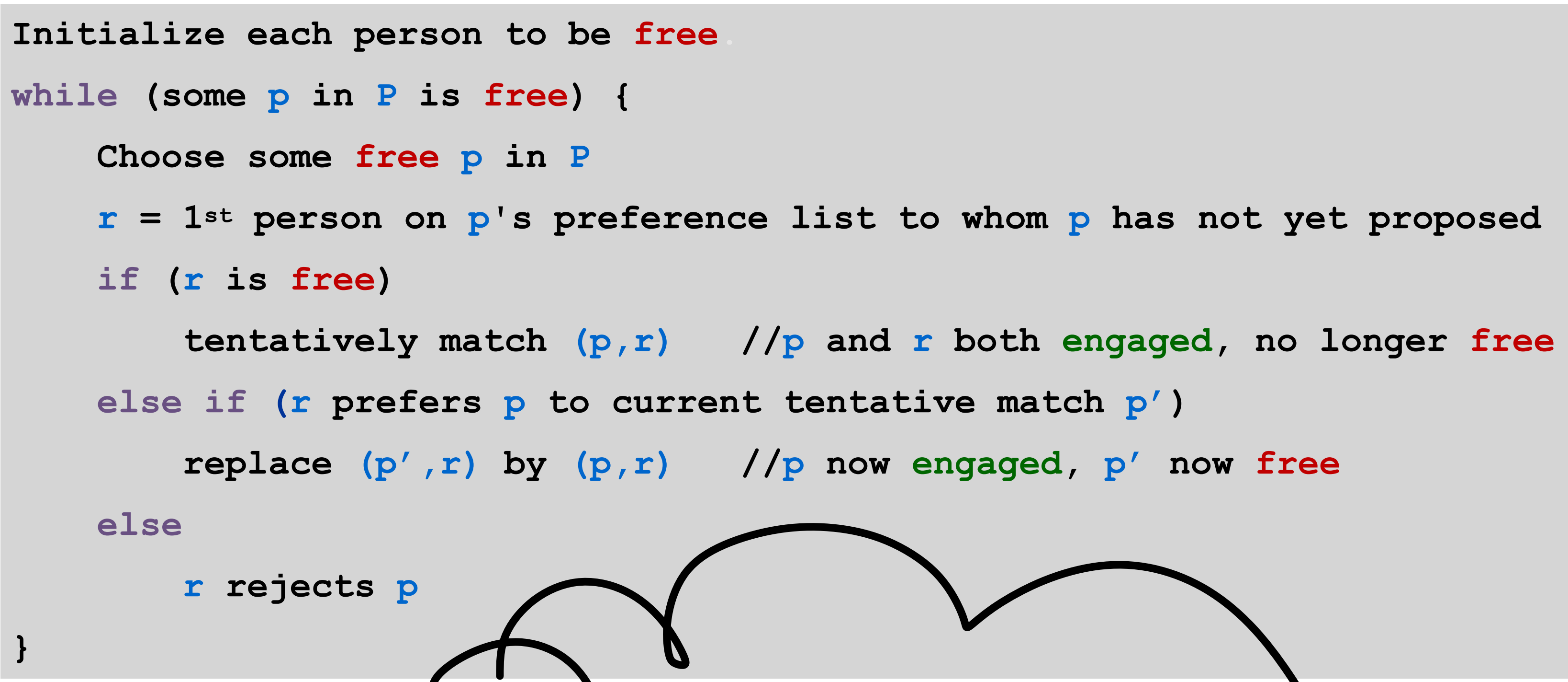
PAPA	F
QUEBEC	B
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA				

	FAV			LEAST
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	Q
CHARLIE	F
DELTA	F

PAPA	F
QUEBEC	B
ROMEO	A
SIERRA	F

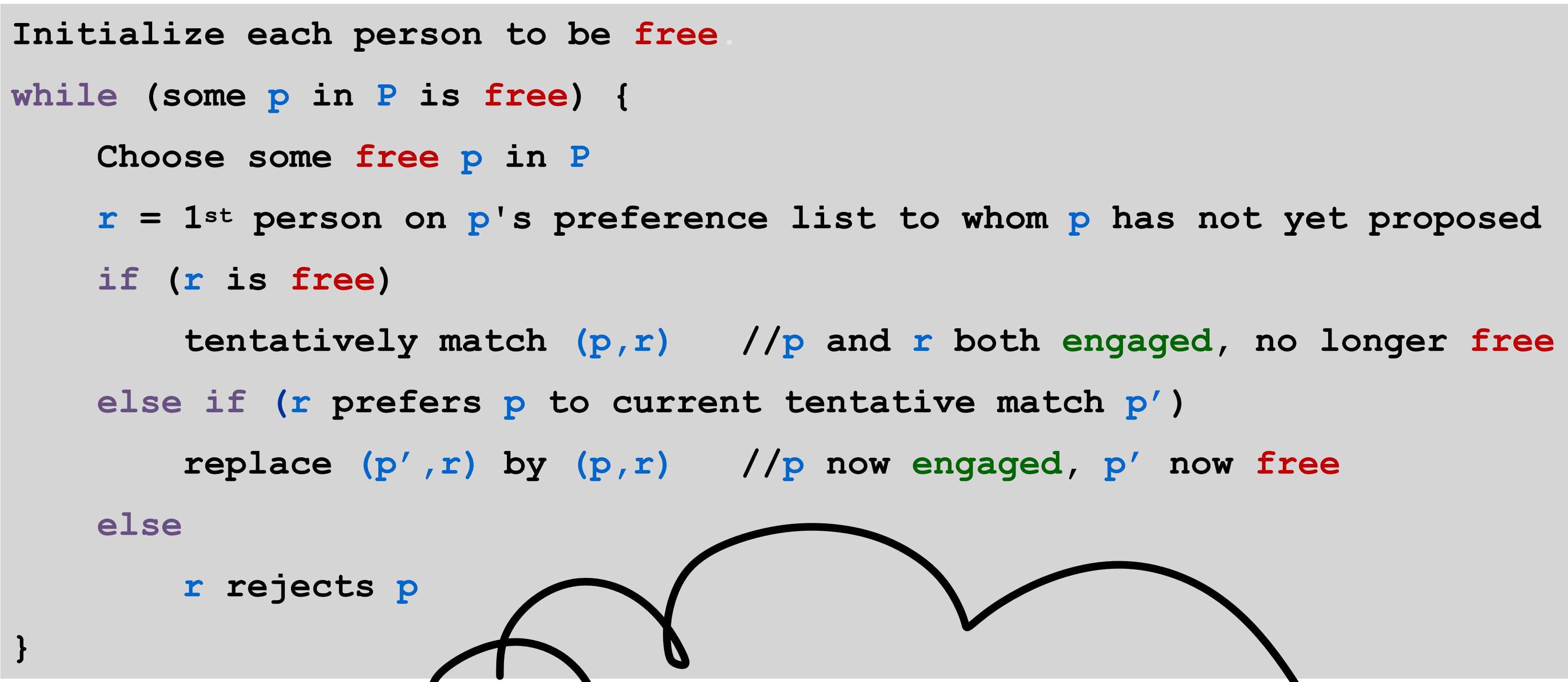
Who do I prefer: Bravo OR Charlie?

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA				

	FAV			LEAST
PAPA				
QUEBEC				
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Who do I prefer: Bravo OR Charlie?

Current partner:

ALPHA	R
BRAVO	Q
CHARLIE	F
DELTA	F

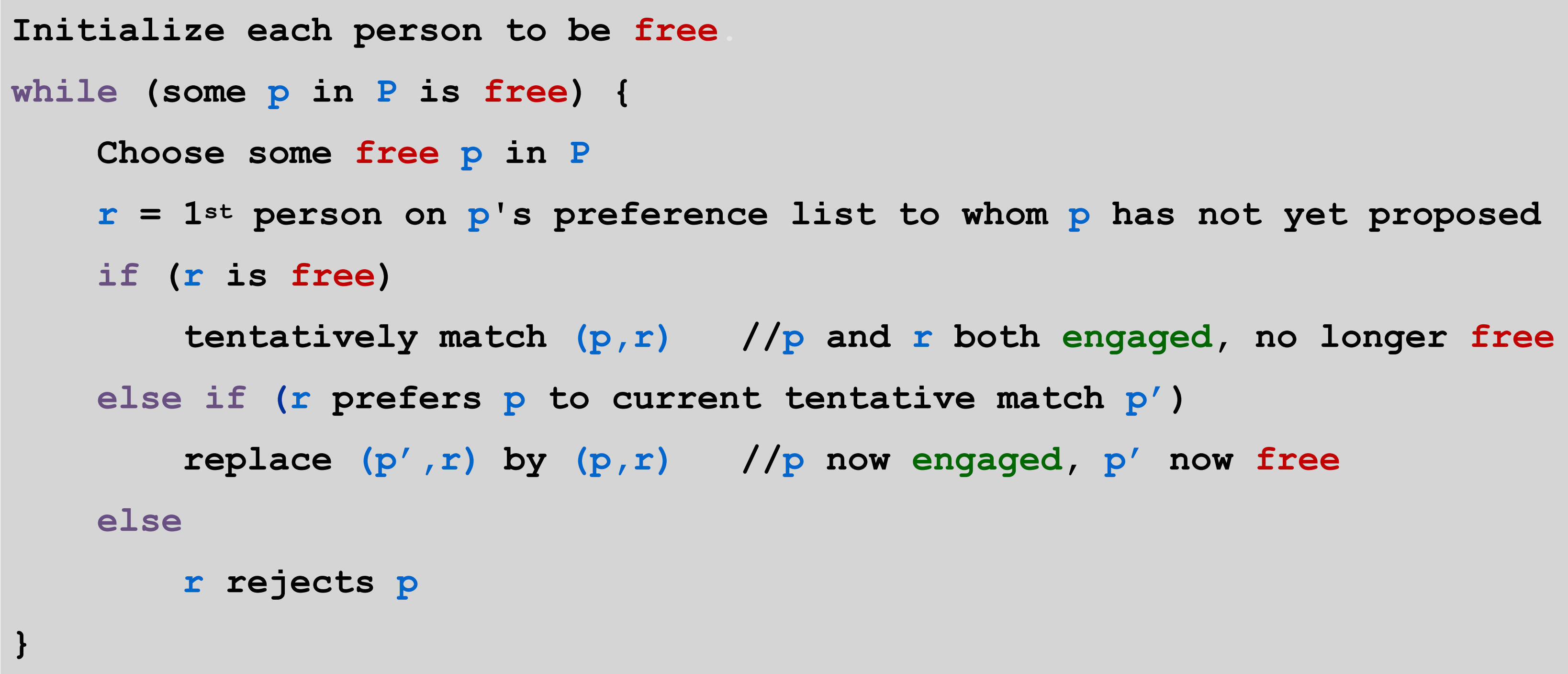
PAPA	F
QUEBEC	B
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA				

	FAV			LEAST
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	F
CHARLIE	Q
DELTA	F

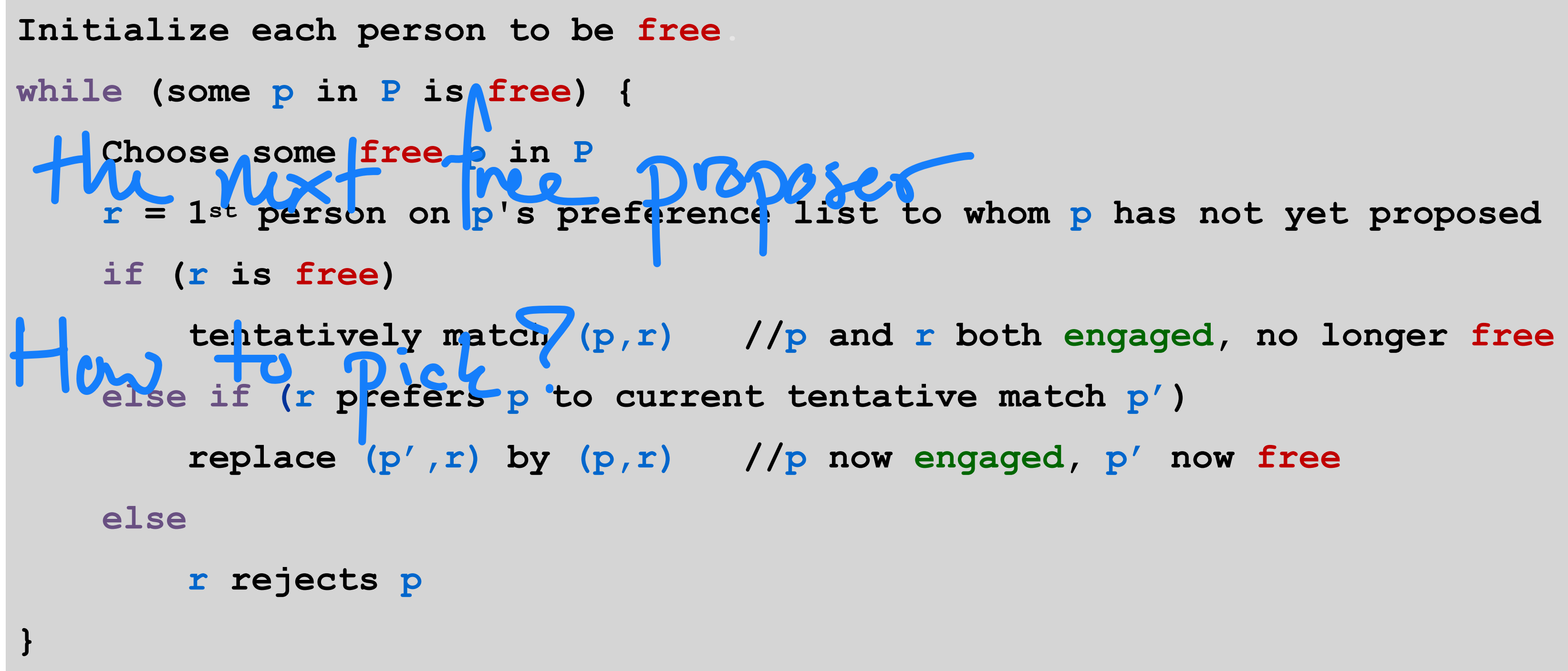
PAPA	F
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA				

	FAV ↓			LEAST ↓
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
   Choose some free p in P
   r = 1st person on p's preference list to whom p has not yet proposed
   if (r is free)
      tentatively match (p,r)    //p and r both engaged, no longer free
   else if (r prefers p to current tentative match p')
      replace (p',r) by (p,r)    //p now engaged, p' now free
   else
      r rejects p
}
```

Pick the next free proposer

How to pick?

Current partner:

ALPHA	R
BRAVO	F
CHARLIE	Q
DELTA	F

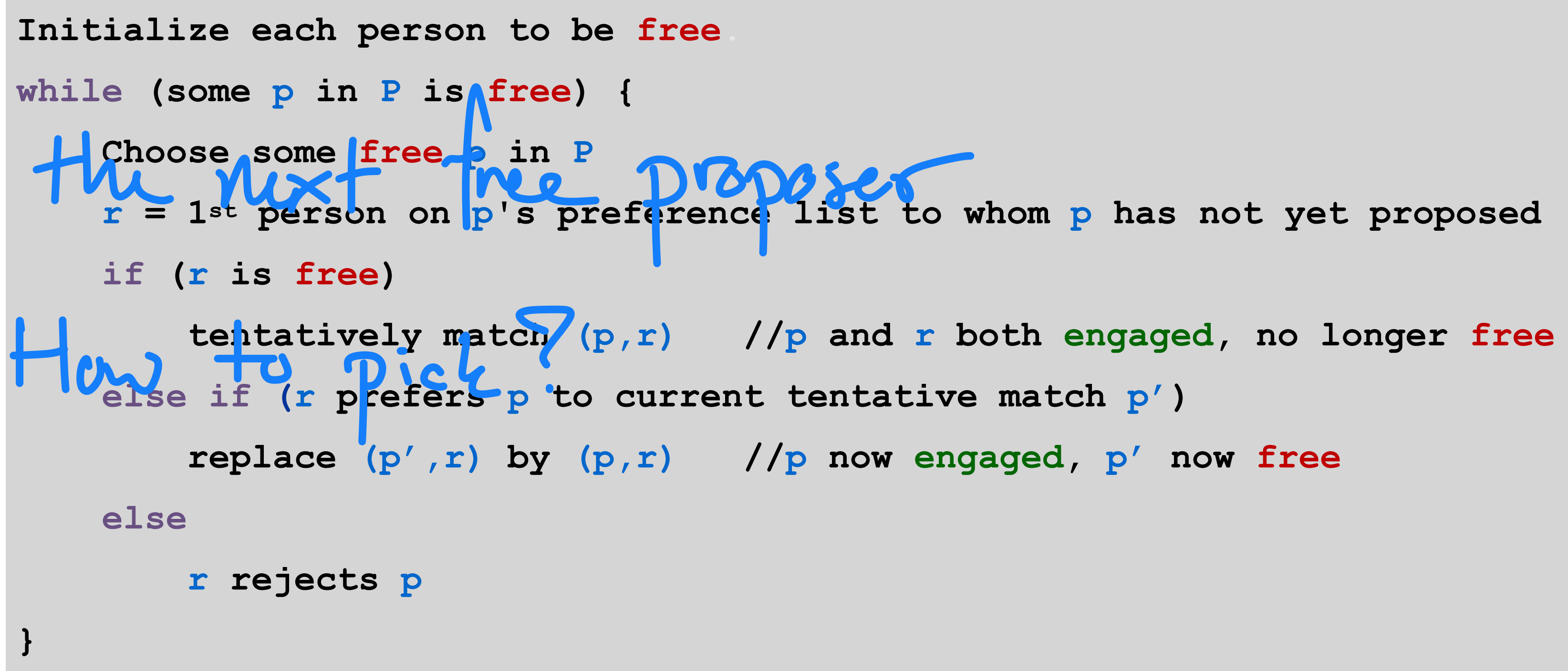
PAPA	F
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA				

	FAV			LEAST
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Pick the next free proposer

How to pick?

Current partner:

ALPHA	R
BRAVO	F
CHARLIE	Q
DELTA	F

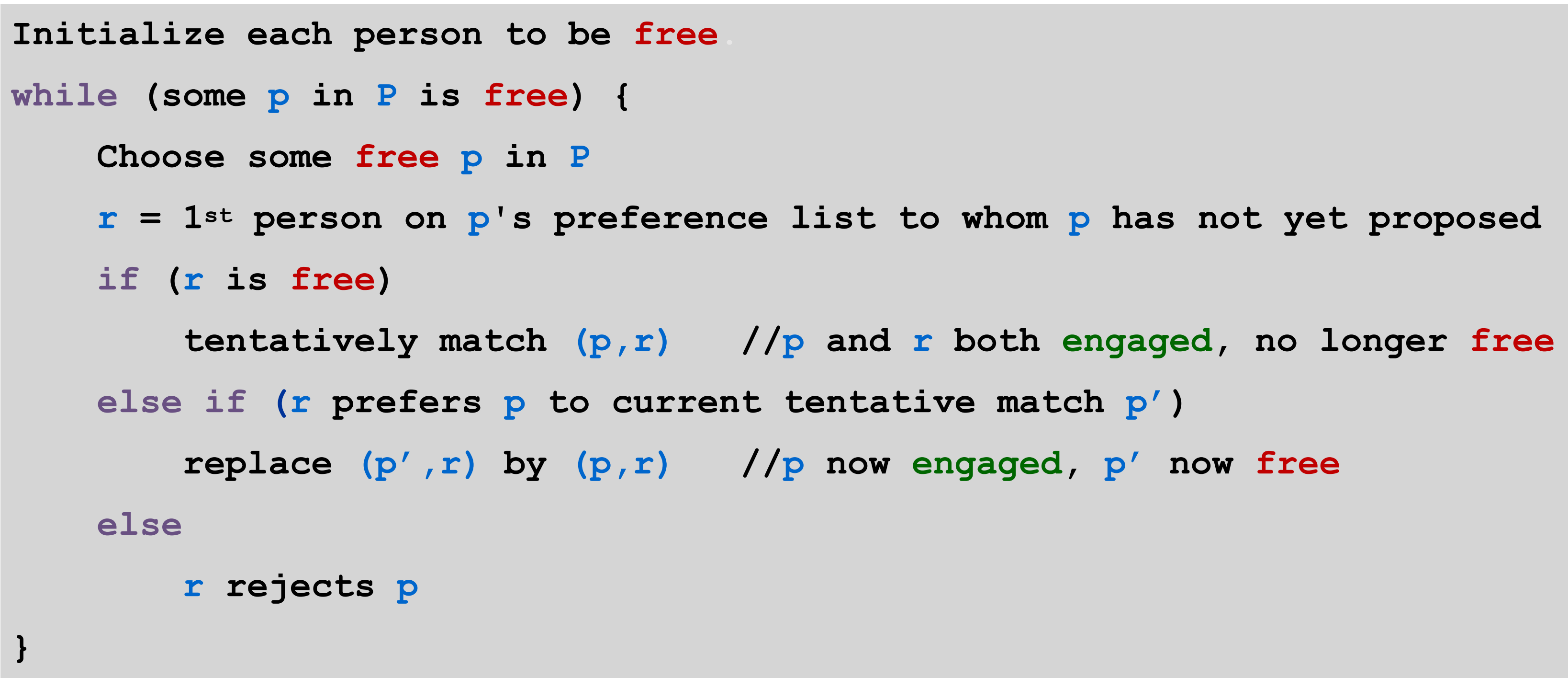
PAPA	F
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA	P			

	FAV			LEAST
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

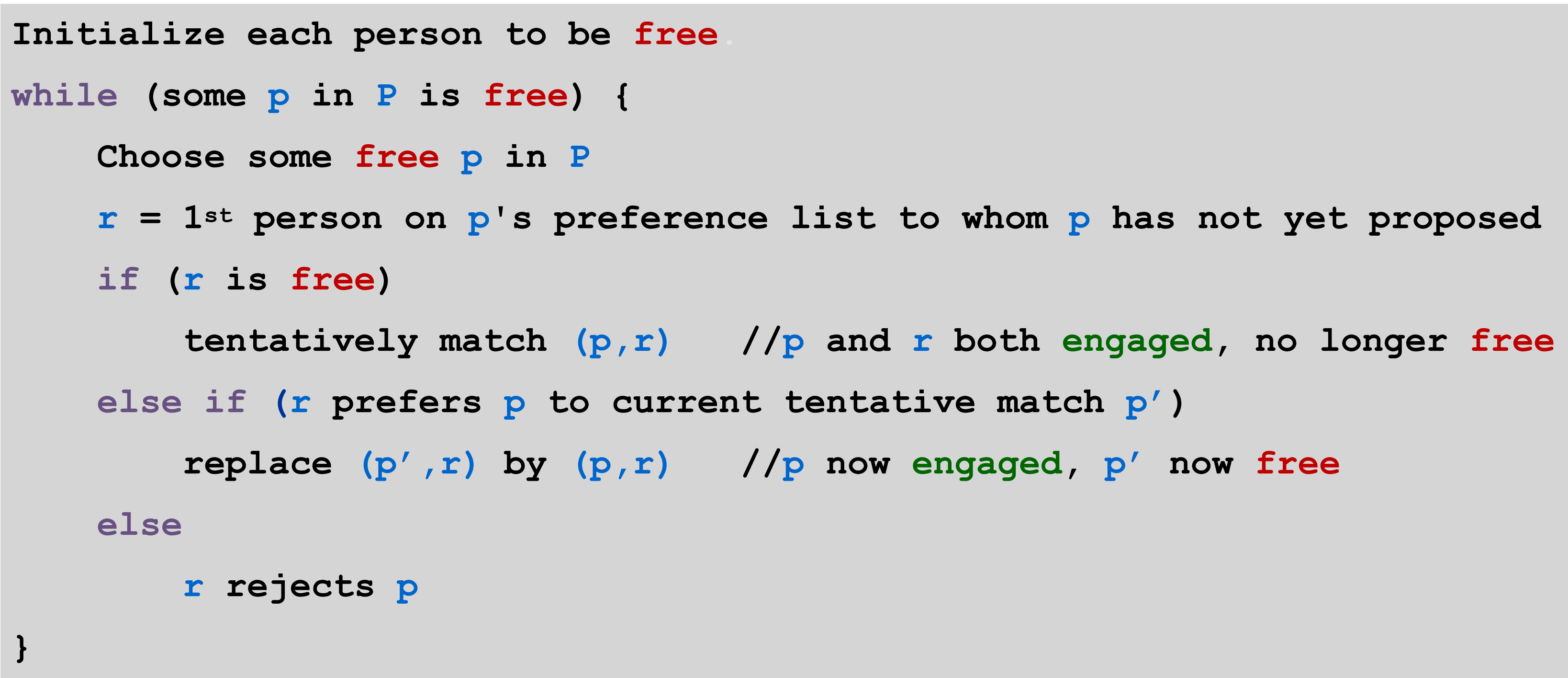
ALPHA	R
BRAVO	F
CHARLIE	Q
DELTA	P

PAPA	P
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q			
CHARLIE	Q			
DELTA	P			

	FAV			LEAST
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Gale-Shapley walkthrough

Current partner:

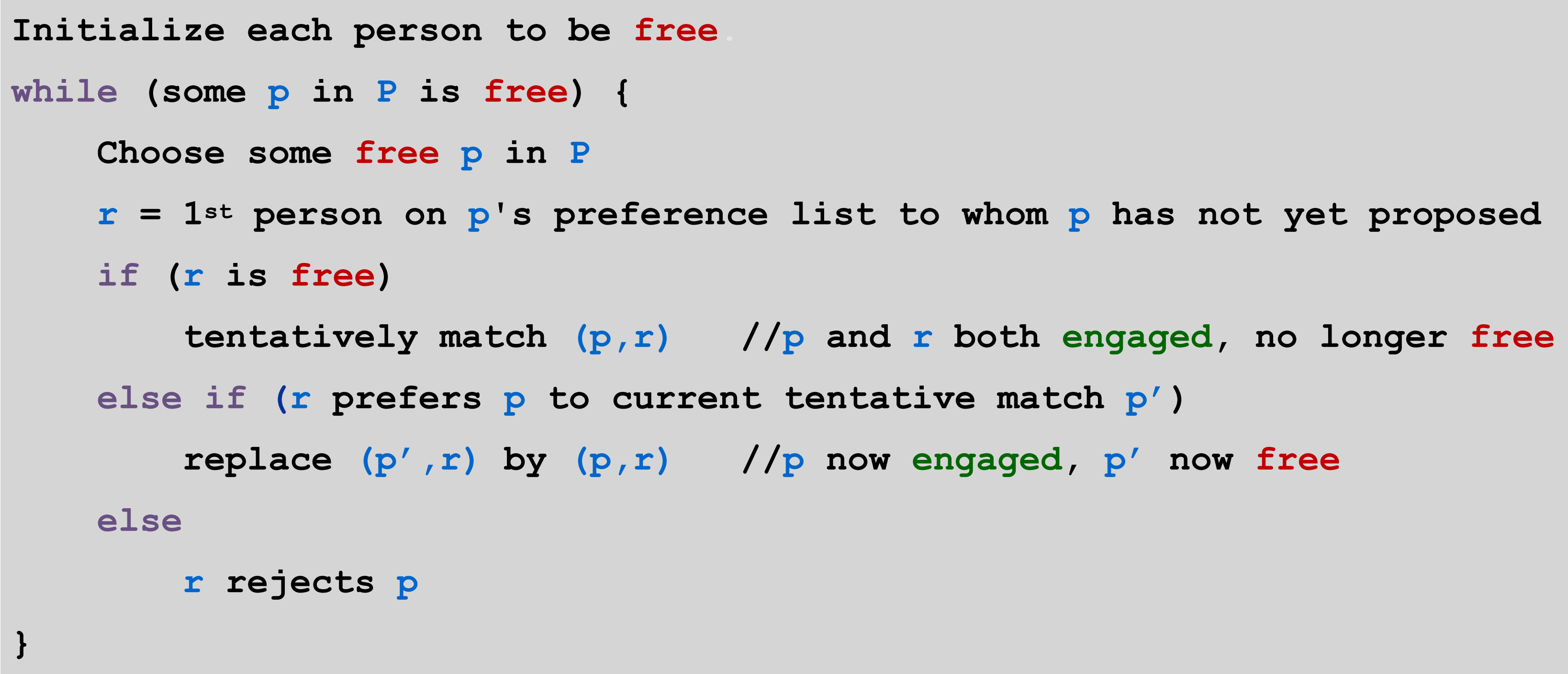
ALPHA	R
BRAVO	F
CHARLIE	Q
DELTA	P

PAPA	P
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO	Q	P		
CHARLIE	Q			
DELTA	P			

	FAV ↓			LEAST ↓
PAPA				
QUEBEC		C		B
ROMEO				
SIERRA				



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Gale-Shapley walkthrough

Current partner:

ALPHA	R
BRAVO	P
CHARLIE	Q
DELTA	F

PAPA	B
QUEBEC	C
ROMEO	A
SIERRA	F

Recievers only trade up

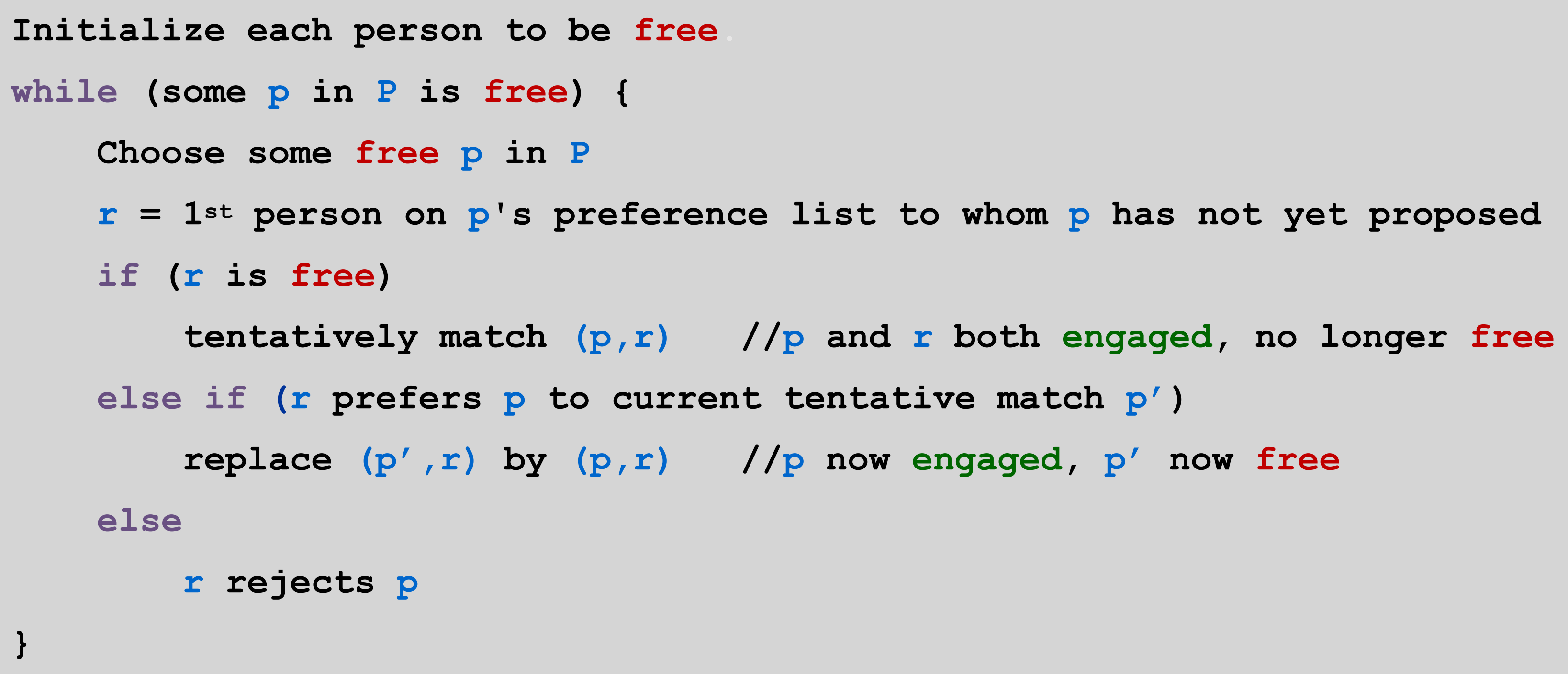
Papa will never change partners again

mark all proposals

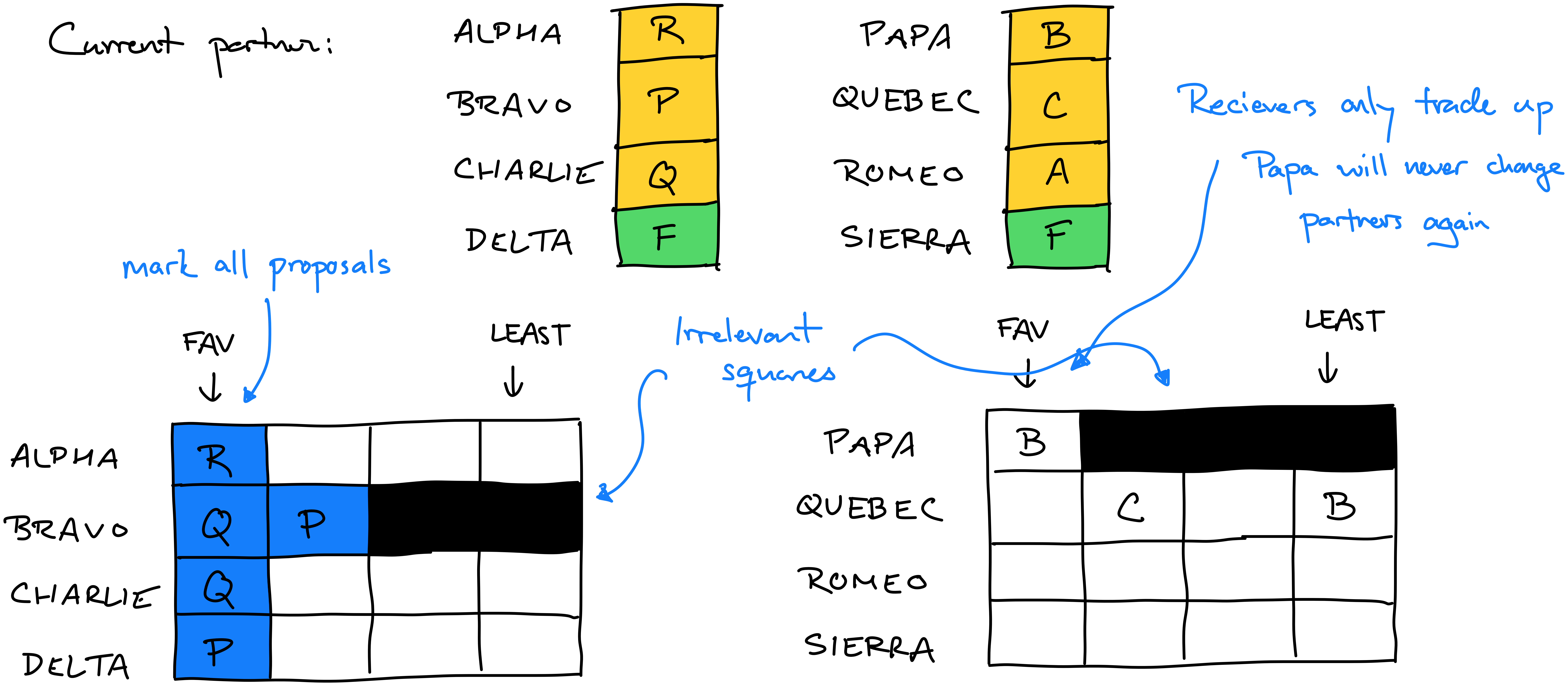
	FAV			LEAST
ALPHA	R			
BRAVO	Q	P		
CHARLIE	Q			
DELTA	P			

	FAV			LEAST
PAPA	B			
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```



Current partner:

ALPHA	R
BRAVO	P
CHARLIE	Q
DELTA	F

PAPA	B
QUEBEC	C
ROMEO	A
SIERRA	F

Recievers only trade up

Papa will never change partners again

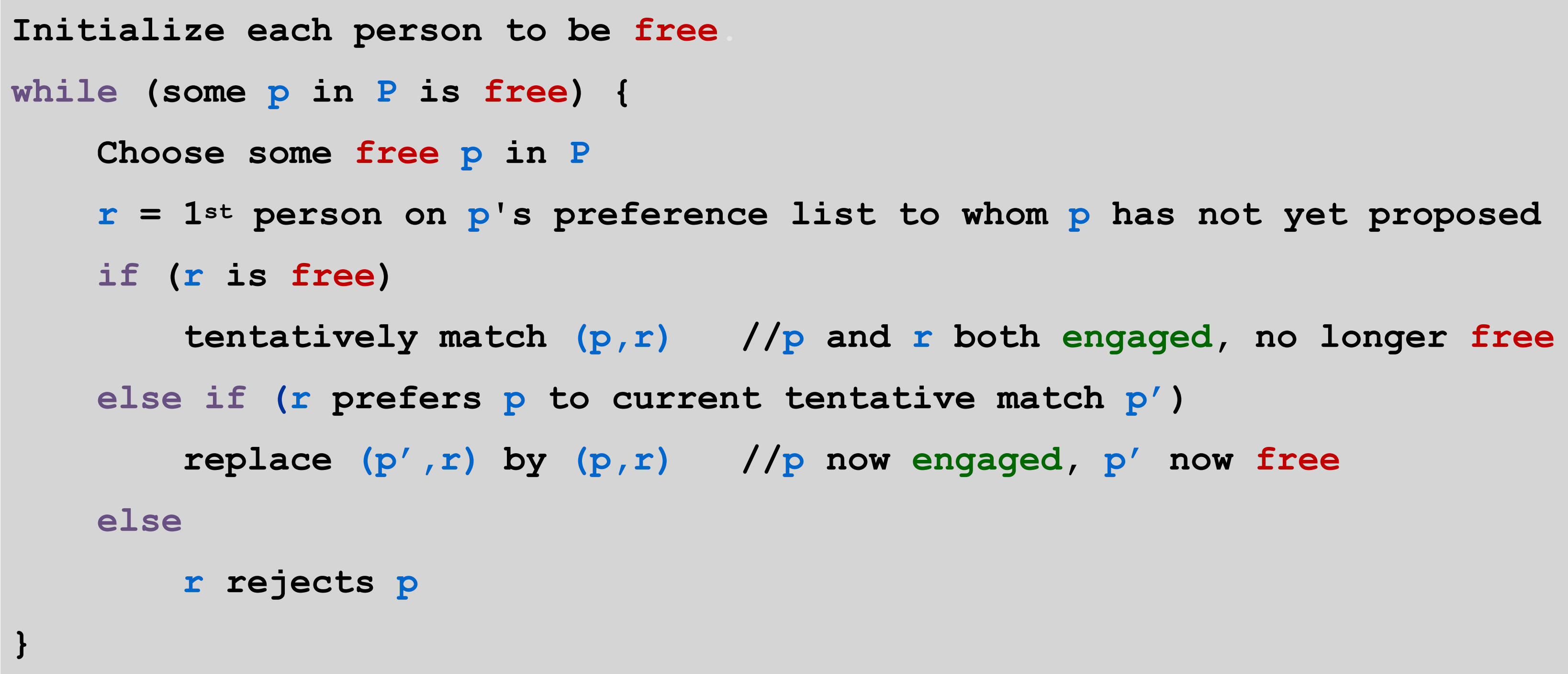
mark all proposals

Irrelevant squares

	FAV			LEAST
ALPHA	R			
BRAVO	Q	P	■	■
CHARLIE	Q			
DELTA	P			

	FAV			LEAST
PAPA	B	■	■	■
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	P
CHARLIE	Q
DELTA	F

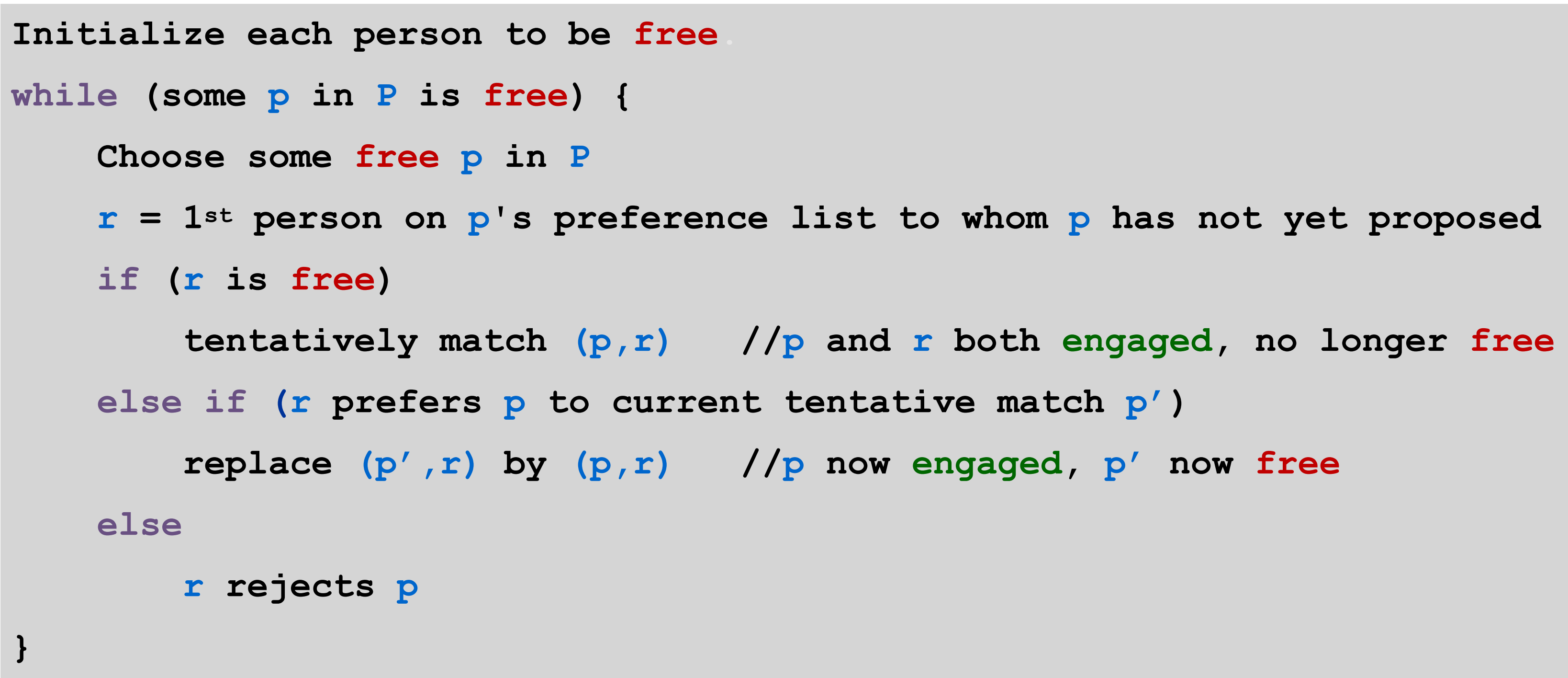
PAPA	B
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO				
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	R
BRAVO	P
CHARLIE	Q
DELTA	F

PAPA	B
QUEBEC	C
ROMEO	A
SIERRA	F

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R			
BRAVO	Q	P		
CHARLIE	Q			
DELTA	P	R		

	FAV ↓			LEAST ↓
PAPA	B			
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough

```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	F
BRAVO	P
CHARLIE	Q
DELTA	R

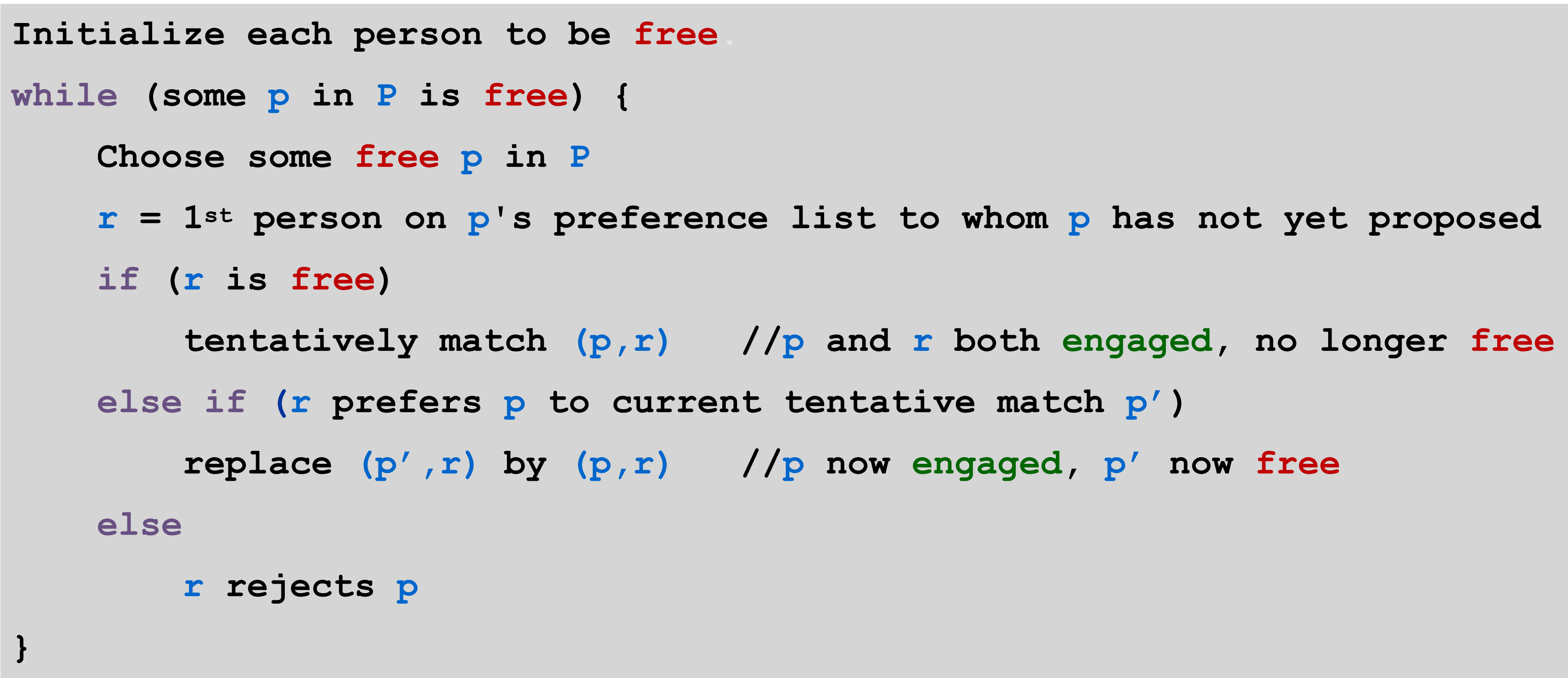
PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R			
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	F
BRAVO	P
CHARLIE	Q
DELTA	R

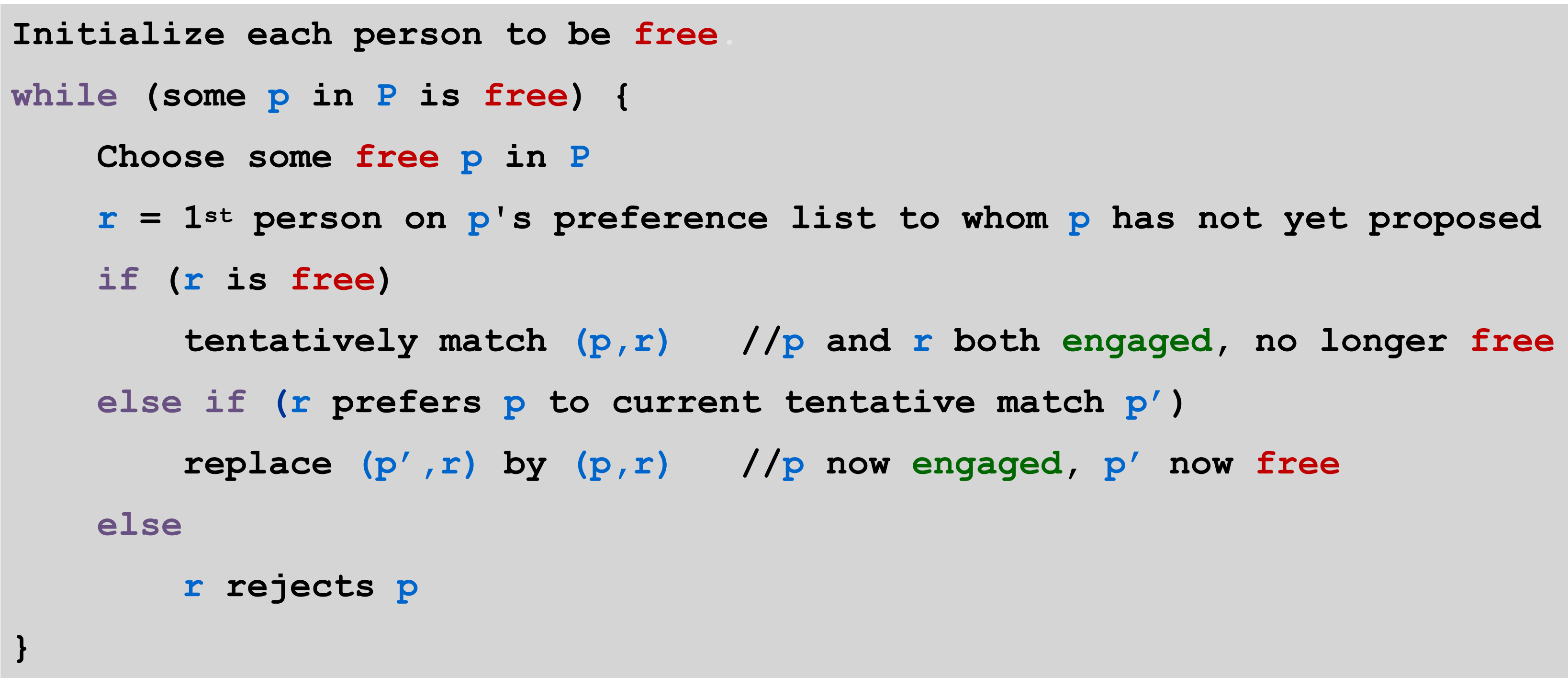
PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	F

mark all proposals

	FAV			LEAST
ALPHA	R	S		
BRAVO	Q	P	■	■
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	■	■	■
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

mark all proposals

	FAV ↓			LEAST ↓
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV ↓			LEAST ↓
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough

no free proposers.

Alg terminates and everyone is matched.

Current partner:

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

mark all proposals

	FAV			LEAST
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

check out how empty the reciever preference matrix is.

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

never even considered

Gale-Shapley walkthrough

no free proposers.

Alg terminates and everyone is matched.

Current partner:

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

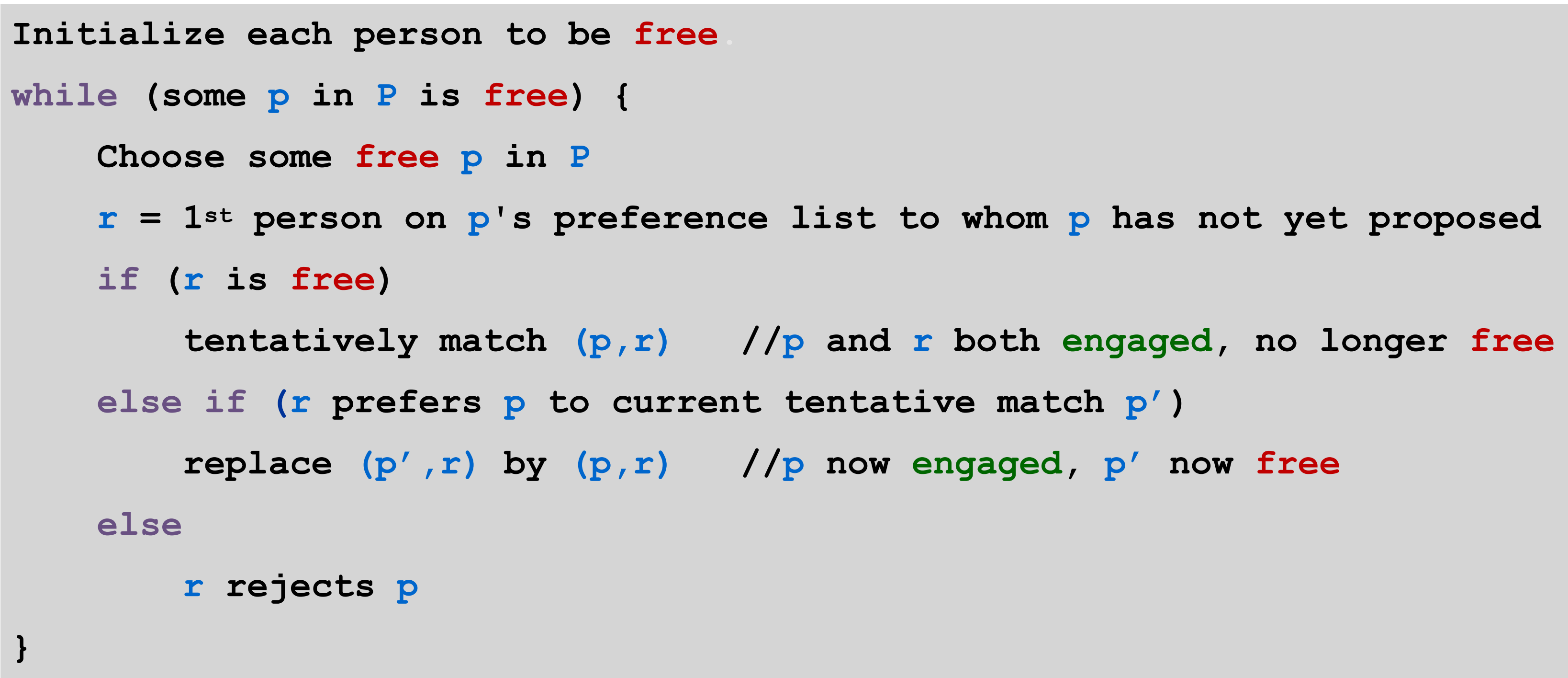
PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

mark all proposals

	FAV			LEAST
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)    //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)    //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

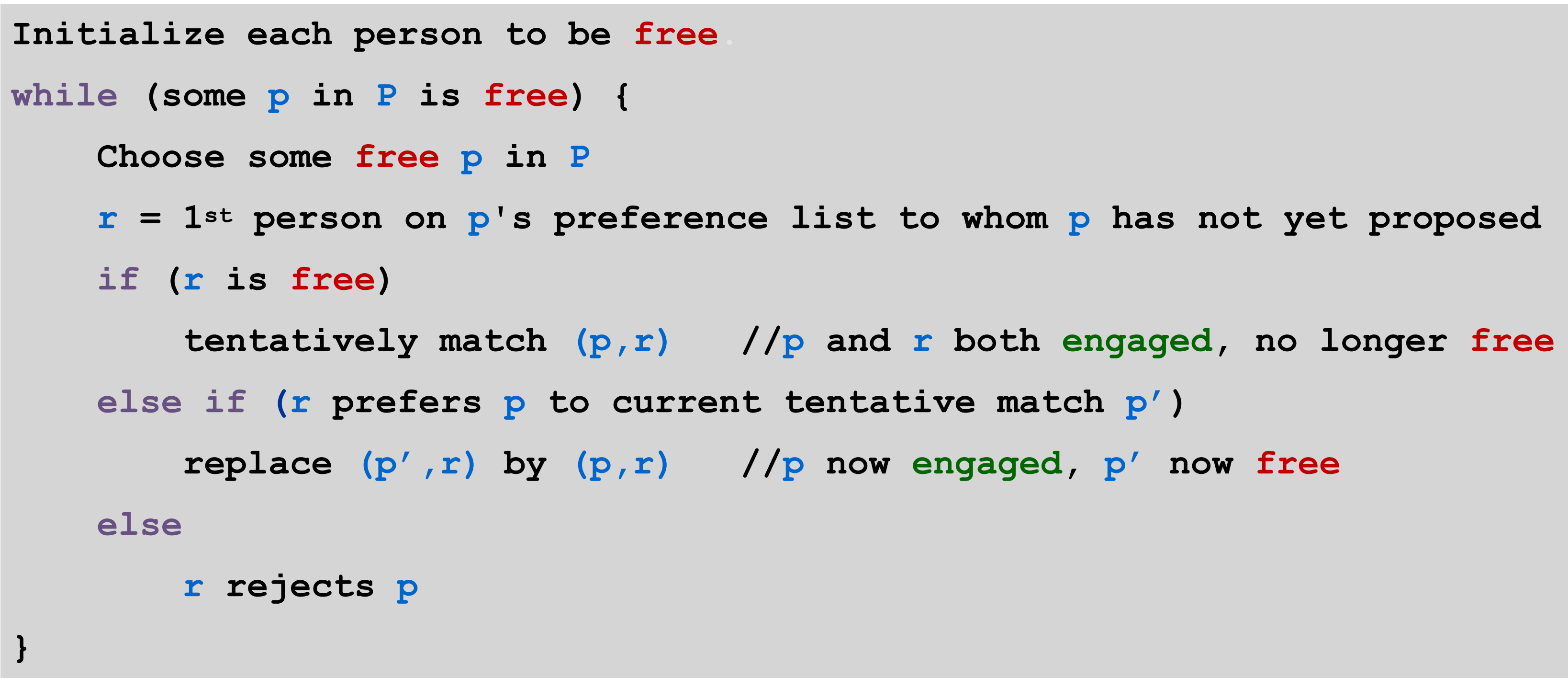
PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

Is (A, R) stable?

	FAV			LEAST
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

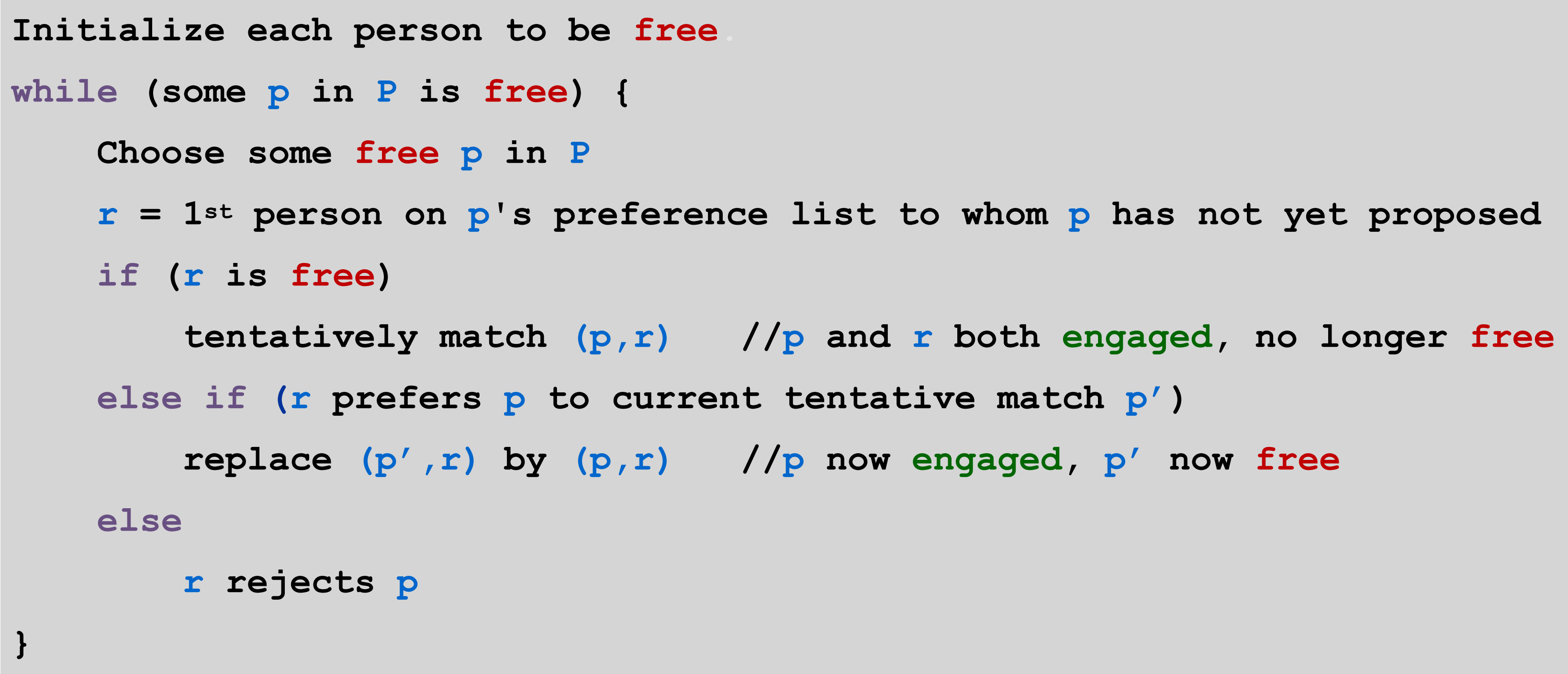
PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

Is (A,Q) stable?

	FAV ↓			LEAST ↓
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV ↓			LEAST ↓
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				

Gale-Shapley walkthrough



```
Initialize each person to be free
while (some p in P is free) {
    Choose some free p in P
    r = 1st person on p's preference list to whom p has not yet proposed
    if (r is free)
        tentatively match (p,r)   //p and r both engaged, no longer free
    else if (r prefers p to current tentative match p')
        replace (p',r) by (p,r)   //p now engaged, p' now free
    else
        r rejects p
}
```

Current partner:

Is (A,P) stable?

ALPHA	S
BRAVO	P
CHARLIE	Q
DELTA	R

PAPA	B
QUEBEC	C
ROMEO	D
SIERRA	A

	FAV			LEAST
ALPHA	R	S		
BRAVO	Q	P	[redacted]	[redacted]
CHARLIE	Q			
DELTA	P	R		

	FAV			LEAST
PAPA	B	[redacted]	[redacted]	[redacted]
QUEBEC		C		B
ROMEO		D	A	
SIERRA				