# Generating Random Derangements

Conrado Martínez[1] Alois Panholzer[2] Helmut Prodinger[3]

[1]Univ. Politècnica de Catalunya, Spain

[2]Tech. Univ. Wien, Austria

[3]Univ. Stellenbosch, South Africa

# Derangements

Le Problème des Derangements:
"A number of gentlemen, say $n$, surrender their top hats in the cloakroom and proceed to the evening's enjoyment. After wining and dining (and wining some more), they stumble back to the cloakroom and confusedly take the first top-hat they see. What is the probability that no gentleman gets his own hat?"

# Derangements

- A derangement is a permutation without fixed points: $\pi(i) \neq i$ for any $i$, $1 \leq i \leq n$
- The number $D_n$ of derangements of size $n$ is

$$D_n = n! \cdot \left[\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + \frac{(-1)^n}{n!}\right] = \left\lfloor \frac{n!+1}{e} \right\rfloor .$$

- As $n \to \infty$, $D_n/n! \sim 1/e \approx 0.36788$.

# Derangements

- A derangement is a permutation without fixed points: $\pi(i) \neq i$ for any $i$, $1 \leq i \leq n$
- The number $D_n$ of derangements of size $n$ is

$$D_n = n! \cdot \left[\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + \frac{(-1)^n}{n!}\right] = \left\lfloor \frac{n!+1}{e} \right\rfloor .$$

- As $n \to \infty$, $D_n/n! \sim 1/e \approx 0.36788$.

# Derangements

- A derangement is a permutation without fixed points: $\pi(i) \neq i$ for any $i$, $1 \leq i \leq n$
- The number $D_n$ of derangements of size $n$ is

$$D_n = n! \cdot \left[ \frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + \frac{(-1)^n}{n!} \right] = \left\lfloor \frac{n!+1}{e} \right\rfloor .$$

- As $n \to \infty$, $D_n / n! \sim 1/e \approx 0.36788$.

# Break #1: Fisher-Yates' shuffle

```
procedure RandomPermutation($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$
    for $i \leftarrow n$ downto 1 do
        $j \leftarrow \text{Uniform}(1, i)$
        $A[i] \leftrightarrow A[j]$
    return $A$
```

# Break #2: Sattolo's algorithm

```
procedure RandomCyclicPermutation($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$
    for $i \leftarrow n$ downto 1 do
        $j \leftarrow \text{Uniform}(1, i - 1)$
        $A[i] \leftrightarrow A[j]$
    return $A$
```

# A recurrence for the number of derangements

$$D_0 = 1, D_1 = 0$$
$$D_n = (n-1)D_{n-1} + (n-1)D_{n-2}$$

# A recurrence for the number of derangements

$$D_0 = 1, D_1 = 0$$
$$D_n = (n-1)D_{n-1} + (n-1)D_{n-2}$$

Choice #1: $n$ belongs to a cycle of length $> 2$.
The derangement of size $n$ is built by constructing a derangement of size $n-1$ and then $n$ is inserted into any of the cycles (of length $\geq 2$); there are $(n-1)$ possible ways to do that

# A recurrence for the number of derangements

$$D_0 = 1, D_1 = 0$$
$$D_n = (n-1)D_{n-1} + (n-1)D_{n-2}$$

Choice #2: $n$ belongs to a cycle of length 2.
The derangement of size $n$ is built by constructing a cycle of size 2 with $n$ and some $j$, $1 \leq j \leq n-1$; then we build a derangement of size $n-2$ with the remaining elements

# The rejection method

```
Require: $n \neq 1$
  procedure RandomDerangement($n$)
      repeat
          $A \leftarrow$ RandomPermutation($n$)
      until Is-Derangement($A$) return $A$
```

$$\mathbb{P}[A \text{ is a derangement}] \approx \frac{1}{e}$$

$$\mathbb{E}[\# \text{ of calls to Random}] = e \cdot n + O(1)$$

# The rejection method

```
Require: $n \neq 1$
  procedure RandomDerangement($n$)
      repeat
          $A \leftarrow$ RandomPermutation($n$)
      until Is-Derangement($A$) return  $A$
```

$$\mathbb{P}[A \text{ is a derangement}] \approx \frac{1}{e}$$

$$\mathbb{E}[\# \text{ of calls to Random}] = e \cdot n + O(1)$$

# The recursive method

```
$C \leftarrow \{1, 2, \ldots, n\}$
RandomDerangement-Rec($n$, $C$)

Require: $n \neq 1$
  procedure RandomDerangement-Rec($n$, $C$)
      if $n \leq 1$ then return
      $j \leftarrow$ a random element from $C$
      $p \leftarrow$ Uniform$(0, 1)$
      if $p < (n-1)D_{n-2}/D_n$ then
          RandomDerangement-Rec($n-2$, $C \setminus \{j, n\}$)
          $\pi(n) \leftarrow j; \pi(j) \leftarrow n$
      else
          RandomDerangement-Rec($n-1$, $C \setminus \{n\}$)
          $\pi(n) \leftarrow \pi(j); \pi(j) \leftarrow n$
```

# Our algorithm

```
Require: $n \neq 1$
  procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
      if $\neg mark[i]$ then
        $j \leftarrow$ pick a random unmarked element in $A[1..i-1]$

        $A[i] \leftrightarrow A[j]$

        if $j$ shall be marked $\equiv$ close the cycle then
          $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
        $u \leftarrow u - 1$
      $i \leftarrow i - 1$
    return $A$
```

# Our algorithm

```
Require: $n \neq 1$
  procedure RandomDerangement($n$)
      for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
      $i \leftarrow n$; $u \leftarrow n$
      while $u \geq 2$ do
          if $\neg mark[i]$ then
              repeat $j \leftarrow \text{Random}(1, i-1)$
              until $\neg mark[j]$
              $A[i] \leftrightarrow A[j]$
              $p \leftarrow \text{Uniform}(0, 1)$
              if $p < (u-1)D_{u-2}/D_u$
                  $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
              $u \leftarrow u - 1$
          $i \leftarrow i - 1$
      return $A$
```

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u-1$
            $u \leftarrow u-1$
        $i \leftarrow i-1$
    return $A$
```

```
1   2   3   4   5   6   7   8
                            ↑
```

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

1 2 3 4 8 6 7 5

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u-1$
            $u \leftarrow u-1$
        $i \leftarrow i-1$
    return $A$
```

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 8 & 6 & 7 & 5 \\ & & & & & & \uparrow & \end{array}$$

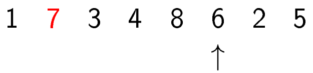
# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

1 7 3 4 8 6 2 5

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```



# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

1 7 3 4 6 8 2 5

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i - 1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u - 1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

```
1   7   3   4   6   8   2   5
                ↑
```

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

1 7 6 4 3 8 2 5

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u - 1$
            $u \leftarrow u - 1$
        $i \leftarrow i - 1$
    return $A$
```

1 7 6 4 3 8 2 5

↑

# Our algorithm

```
procedure RandomDerangement($n$)
    for $i \leftarrow 1$ to $n$ do $A[i] \leftarrow i$; $mark[i] \leftarrow$ false
    $i \leftarrow n$; $u \leftarrow n$
    while $u \geq 2$ do
        if $\neg mark[i]$ then
            repeat $j \leftarrow$ Random$(1, i-1)$
            until $\neg mark[j]$
            $A[i] \leftrightarrow A[j]$
            $p \leftarrow$ Uniform$(0, 1)$
            if $p < (u-1)D_{u-2}/D_u$ then
                $mark[j] \leftarrow$ true; $u \leftarrow u-1$
            $u \leftarrow u-1$
        $i \leftarrow i-1$
    return $A$
```

4 7 6 1 3 8 2 5

# The analysis

- # of marked elements = # of cycles ($C_n$)
- # of iterations = # of calls to Uniform = $n - C_n$
- $G$ = # of calls to Random
- $G_i$ = # of calls to Random at iteration $i$
-

$$\begin{aligned} \mathbb{E}[\text{cost}] &= n - \mathbb{E}[C_n] + \mathbb{E}[G] \\ &= n - \mathbb{E}[C_n] + \sum_{1<i\leq n} \mathbb{E}[G_i] \end{aligned}$$

# The analysis

- # of marked elements = # of cycles ($C_n$)
- # of iterations = # of calls to Uniform = $n - C_n$
- $G$ = # of calls to Random
- $G_i$ = # of calls to Random at iteration $i$
-

$$\begin{aligned}\mathbb{E}[\text{cost}] &= n - \mathbb{E}[C_n] + \mathbb{E}[G] \\ &= n - \mathbb{E}[C_n] + \sum_{1<i\leq n} \mathbb{E}[G_i]\end{aligned}$$

# The analysis

- # of marked elements = # of cycles ($C_n$)
- # of iterations = # of calls to Uniform = $n - C_n$
- $G$ = # of calls to Random
- $G_i$ = # of calls to Random at iteration $i$
-

$$\begin{aligned}\mathbb{E}[\text{cost}] &= n - \mathbb{E}[C_n] + \mathbb{E}[G] \\ &= n - \mathbb{E}[C_n] + \sum_{1<i\leq n} \mathbb{E}[G_i]\end{aligned}$$

# The analysis

- # of marked elements = # of cycles ($C_n$)
- # of iterations = # of calls to Uniform = $n - C_n$
- $G$ = # of calls to Random
- $G_i$ = # of calls to Random at iteration $i$
-

$$\begin{aligned}\mathbb{E}[\text{cost}] &= n - \mathbb{E}[C_n] + \mathbb{E}[G] \\ &= n - \mathbb{E}[C_n] + \sum_{1<i\leq n} \mathbb{E}[G_i]\end{aligned}$$

# The analysis

- # of marked elements = # of cycles ($C_n$)
- # of iterations = # of calls to Uniform = $n - C_n$
- $G$ = # of calls to Random
- $G_i$ = # of calls to Random at iteration $i$
-

$$\begin{aligned}\mathbb{E}[\text{cost}] &= n - \mathbb{E}[C_n] + \mathbb{E}[G] \\ &= n - \mathbb{E}[C_n] + \sum_{1<i\leq n} \mathbb{E}[G_i]\end{aligned}$$

# The analysis

The computation of $\mathbb{E}[C_n]$ can be done via standard generating function techniques:

$$C(z,v) = \sum_{A\in\mathcal{D}} \frac{z^{|A|}}{|A|!} v^{\#\text{ cycles}(A)}$$

$$= \exp\left(v\left(\log\frac{1}{1-z} - z\right)\right) = e^{-vz}\frac{1}{(1-z)^{v}}$$

$$\mathbb{E}\left[v^{C_n}\right] = \frac{n!}{D_n}[z^n]C(z,v) = \frac{e^{1-v}}{(v-1)!}n^{v-1}(1+O(n^{-1+\epsilon}))$$

$$\mathbb{E}[C_n] = \log n + O(1), \quad \mathbb{V}[C_n] = \log n + O(1)$$

$$\frac{C_n - \log n}{\sqrt{\log n}} \to \mathcal{N}(0,1)$$

# The analysis

The computation of $\mathbb{E}[C_n]$ can be done via standard generating function techniques:

$$
\begin{aligned}
C(z,v) &= \sum_{A \in \mathcal{D}} \frac{z^{|A|}}{|A|!} v^{\#\text{ cycles}(A)} \\
&= \exp\left(v\left(\log\frac{1}{1-z} - z\right)\right) = e^{-vz}\frac{1}{(1-z)^v}
\end{aligned}
$$

$$
\mathbb{E}\left[v^{C_n}\right] = \frac{n!}{D_n}[z^n]C(z,v) = \frac{e^{1-v}}{(v-1)!}n^{v-1}(1+O(n^{-1+\epsilon}))
$$

$$
\mathbb{E}[C_n] = \log n + O(1), \quad \mathbb{V}[C_n] = \log n + O(1)
$$

$$
\frac{C_n - \log n}{\sqrt{\log n}} \to \mathcal{N}(0,1)
$$

# The analysis

The computation of $\mathbb{E}[C_n]$ can be done via standard generating function techniques:

$$C(z,v) = \sum_{A \in \mathcal{D}} \frac{z^{|A|}}{|A|!} v^{\#\ \mathrm{cycles}(A)}$$

$$= \exp\left(v\left(\log\frac{1}{1-z} - z\right)\right) = e^{-vz}\frac{1}{(1-z)^v}$$

$$\mathbb{E}\left[v^{C_n}\right] = \frac{n!}{D_n}[z^n]C(z,v) = \frac{e^{1-v}}{(v-1)!}n^{v-1}(1+O(n^{-1+\epsilon}))$$

$$\mathbb{E}[C_n] = \log n + O(1), \quad \mathbb{V}[C_n] = \log n + O(1)$$

$$\frac{C_n - \log n}{\sqrt{\log n}} \to \mathcal{N}(0,1)$$

# The analysis

The computation of $\mathbb{E}[C_n]$ can be done via standard generating function techniques:

$$C(z,v) = \sum_{A \in \mathcal{D}} \frac{z^{|A|}}{|A|!} v^{\#\text{ cycles}(A)}$$

$$= \exp\left(v\left(\log\frac{1}{1-z} - z\right)\right) = e^{-vz}\frac{1}{(1-z)^v}$$

$$\mathbb{E}\left[v^{C_n}\right] = \frac{n!}{D_n}[z^n]C(z,v) = \frac{e^{1-v}}{(v-1)!}n^{v-1}(1+O(n^{-1+\epsilon}))$$

$$\mathbb{E}[C_n] = \log n + O(1), \quad \mathbb{V}[C_n] = \log n + O(1)$$

$$\frac{C_n - \log n}{\sqrt{\log n}} \to \mathcal{N}(0,1)$$

# The analysis

The computation of $\mathbb{E}[C_n]$ can be done via standard generating function techniques:

$$\begin{aligned} C(z,v) &= \sum_{A\in\mathcal{D}} \frac{z^{|A|}}{|A|!} v^{\#\ \text{cycles}(A)} \\ &= \exp\left(v\left(\log\frac{1}{1-z} - z\right)\right) = e^{-vz}\frac{1}{(1-z)^v} \\ \mathbb{E}\left[v^{C_n}\right] &= \frac{n!}{D_n}[z^n]C(z,v) = \frac{e^{1-v}}{(v-1)!}n^{v-1}(1+O(n^{-1+\epsilon})) \\ \mathbb{E}[C_n] &= \log n + O(1), \quad \mathbb{V}[C_n] = \log n + O(1) \\ \frac{C_n - \log n}{\sqrt{\log n}} &\to \mathcal{N}(0,1) \end{aligned}$$

# The analysis

- $M_i$ indicator variable for the event "$A[i]$ gets marked"
- $M_i = 1 \implies G_i = 0$
-

$$\mathbb{E}[G] = \sum_{1<i\leq n} \mathbb{E}[G_i \mid M_i = 0] \cdot \mathbb{P}[M_i = 0]$$

# The analysis

- $M_i$ indicator variable for the event "$A[i]$ gets marked"
- $M_i = 1 \implies G_i = 0$
- $$\mathbb{E}[G] = \sum_{1<i\leq n} \mathbb{E}[G_i \mid M_i = 0] \cdot \mathbb{P}[M_i = 0]$$

# The analysis

- $M_i$ indicator variable for the event "$A[i]$ gets marked"
- $M_i = 1 \implies G_i = 0$
-

$$\mathbb{E}[G] = \sum_{1 < i \leq n} \mathbb{E}[G_i \mid M_i = 0] \cdot \mathbb{P}[M_i = 0]$$

# The analysis

- $U_i = \#$ of unmarked elements in $A[1..i]$; $U_n = n$
- $B_{i+1} = \#$ of marked elements in $A[1..i]$; $B_{n+1} = 0$
- $U_i + B_{i+1} = i$
- If $A[i]$ is not marked then $G_i$ is geometrically distributed with probability of success $(U_i - 1)/(i-1) = (i - 1 - B_{i+1})/(i-1)$; hence

$$\mathbb{E}[G_i \mid M_i = 0] = \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right]$$

# The analysis

- $U_i = \#$ of unmarked elements in $A[1..i]$; $U_n = n$
- $B_{i+1} = \#$ of marked elements in $A[1..i]$; $B_{n+1} = 0$
- $U_i + B_{i+1} = i$
- If $A[i]$ is not marked then $G_i$ is geometrically distributed with probability of success $(U_i - 1)/(i-1) = (i-1-B_{i+1})/(i-1)$; hence

$$\mathbb{E}[G_i \mid M_i = 0] = \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right]$$

# The analysis

- $U_i = \#$ of unmarked elements in $A[1..i]$; $U_n = n$
- $B_{i+1} = \#$ of marked elements in $A[1..i]$; $B_{n+1} = 0$
- $U_i + B_{i+1} = i$
- If $A[i]$ is not marked then $G_i$ is geometrically distributed with probability of success $(U_i - 1)/(i-1) = (i - 1 - B_{i+1})/(i-1)$; hence

$$\mathbb{E}[G_i \mid M_i = 0] = \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right]$$

# The analysis

- $U_i = \#$ of unmarked elements in $A[1..i]$; $U_n = n$
- $B_{i+1} = \#$ of marked elements in $A[1..i]$; $B_{n+1} = 0$
- $U_i + B_{i+1} = i$
- If $A[i]$ is not marked then $G_i$ is geometrically distributed with probability of success $(U_i - 1)/(i-1) = (i - 1 - B_{i+1})/(i-1)$; hence

$$\mathbb{E}[G_i \mid M_i = 0] = \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right]$$

# The analysis

- $B_{i+1} \leq C_n$
- $0 \leq B_{i+1} \leq i$
- $U_i \neq 1$ and $B_{i+1} \neq 1$ for all $1 \leq i \leq n$
- If $M_i = 0$ then $B_{i+1} < i - 1$

# The analysis

- $B_{i+1} \leq C_n$
- $0 \leq B_{i+1} \leq i$
- $U_i \neq 1$ and $B_{i+1} \neq 1$ for all $1 \leq i \leq n$
- If $M_i = 0$ then $B_{i+1} < i - 1$

# The analysis

- $B_{i+1} \leq C_n$
- $0 \leq B_{i+1} \leq i$
- $U_i \neq 1$ and $B_{i+1} \neq 1$ for all $1 \leq i \leq n$
- If $M_i = 0$ then $B_{i+1} < i - 1$

# The analysis

- $B_{i+1} \leq C_n$
- $0 \leq B_{i+1} \leq i$
- $U_i \neq 1$ and $B_{i+1} \neq 1$ for all $1 \leq i \leq n$
- If $M_i = 0$ then $B_{i+1} < i - 1$

# The analysis

$$
\begin{aligned}
\mathbb{E}[G] &= \sum_{1<i\le n} \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right] \cdot \mathbb{P}[M_i = 0] \\
&\le \sum_{1<i\le n} \mathbb{E}\left[\min\left\{i-1, \frac{i-1}{i-1-C_n}\right\}\right] \\
&\le \sum_{1\le k\le \lfloor n/2\rfloor} \mathbb{P}[C_n = k] \left(\sum_{i=1}^{k+1}(i-1) + \sum_{i=k+2}^{\lfloor n/2\rfloor} \frac{i-1}{i-1-k}\right) \\
&= n - 1 - \mathbb{E}[C_n] + \frac{1}{2}\,\mathbb{E}\left[C_n^2\right] + O(\mathbb{E}[C_n \log(n - C_n)]) \\
&= n + O(\mathbb{E}\left[C_n^2\right]) + O(\log n \cdot \mathbb{E}[C_n]) = n + O(\log^2 n)
\end{aligned}
$$

# The analysis

$$
\begin{aligned}
\mathbb{E}[G] &= \sum_{1<i\leq n} \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right] \cdot \mathbb{P}[M_i = 0] \\
&\leq \sum_{1<i\leq n} \mathbb{E}\left[\min\left\{i-1, \frac{i-1}{i-1-C_n}\right\}\right] \\
&\leq \sum_{1\leq k\leq \lfloor n/2\rfloor} \mathbb{P}[C_n = k] \left(\sum_{i=1}^{k+1}(i-1) + \sum_{i=k+2}^{\lfloor n/2\rfloor} \frac{i-1}{i-1-k}\right) \\
&= n - 1 - \mathbb{E}[C_n] + \frac{1}{2}\,\mathbb{E}\left[C_n^2\right] + O(\mathbb{E}[C_n \log(n - C_n)]) \\
&= n + O(\mathbb{E}\left[C_n^2\right]) + O(\log n \cdot \mathbb{E}[C_n]) = n + O(\log^2 n)
\end{aligned}
$$

# The analysis

$$
\begin{aligned}
\mathbb{E}[G] &= \sum_{1<i\leq n} \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right] \cdot \mathbb{P}[M_i = 0] \\
&\leq \sum_{1<i\leq n} \mathbb{E}\left[\min\left\{i-1, \frac{i-1}{i-1-C_n}\right\}\right] \\
&\leq \sum_{1\leq k\leq \lfloor n/2\rfloor} \mathbb{P}[C_n = k] \left(\sum_{i=1}^{k+1}(i-1) + \sum_{i=k+2}^{\lfloor n/2\rfloor} \frac{i-1}{i-1-k}\right) \\
&= n - 1 - \mathbb{E}[C_n] + \frac{1}{2}\,\mathbb{E}\left[C_n^2\right] + O(\mathbb{E}[C_n \log(n - C_n)]) \\
&= n + O(\mathbb{E}\left[C_n^2\right]) + O(\log n \cdot \mathbb{E}[C_n]) = n + O(\log^2 n)
\end{aligned}
$$

# The analysis

$$
\begin{aligned}
\mathbb{E}[G] &= \sum_{1<i\leq n} \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right] \cdot \mathbb{P}[M_i = 0] \\
&\leq \sum_{1<i\leq n} \mathbb{E}\left[\min\left\{i-1, \frac{i-1}{i-1-C_n}\right\}\right] \\
&\leq \sum_{1\leq k\leq \lfloor n/2\rfloor} \mathbb{P}[C_n = k]\left(\sum_{i=1}^{k+1}(i-1) + \sum_{i=k+2}^{\lfloor n/2\rfloor} \frac{i-1}{i-1-k}\right) \\
&= n - 1 - \mathbb{E}[C_n] + \frac{1}{2}\,\mathbb{E}\left[C_n^2\right] + O(\mathbb{E}[C_n \log(n - C_n)]) \\
&= n + O(\mathbb{E}\left[C_n^2\right]) + O(\log n \cdot \mathbb{E}[C_n]) = n + O(\log^2 n)
\end{aligned}
$$

# The analysis

$$
\begin{aligned}
\mathbb{E}[G] &= \sum_{1<i\leq n} \mathbb{E}\left[\frac{i-1}{i-1-B_{i+1}} \mid M_i = 0\right] \cdot \mathbb{P}[M_i = 0] \\
&\leq \sum_{1<i\leq n} \mathbb{E}\left[\min\left\{i-1, \frac{i-1}{i-1-C_n}\right\}\right] \\
&\leq \sum_{1\leq k\leq \lfloor n/2 \rfloor} \mathbb{P}[C_n = k] \left(\sum_{i=1}^{k+1}(i-1) + \sum_{i=k+2}^{\lfloor n/2 \rfloor} \frac{i-1}{i-1-k}\right) \\
&= n - 1 - \mathbb{E}[C_n] + \frac{1}{2}\,\mathbb{E}\big[C_n^2\big] + O(\mathbb{E}[C_n \log(n - C_n)]) \\
&= n + O(\mathbb{E}\big[C_n^2\big]) + O(\log n \cdot \mathbb{E}[C_n]) = n + O(\log^2 n)
\end{aligned}
$$

# The analysis

Since we also have $\mathbb{E}[G] \geq n - \mathbb{E}[C_n]$, we have finally

$$\mathbb{E}[\text{cost}] = n - \mathbb{E}[C_n] + \mathbb{E}[G] = 2n + O(\log^2 n)$$