Analysis of Algorithms Probabilistic Analysis

Andres Mendez-Vazquez

January 10, 2018

イロン イボン イヨン トヨ

1/27

Outline

The Hiring ProblemIntroduction

2 The Hiring Algorithm

- An Initial Algorithm
- What do we want?
- Indicator Random Variable
 - The Indicator Function

4 The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

Outline

The Hiring ProblemIntroduction

2 The Hiring Algorithm

- An Initial Algorithm
- What do we want?
- Indicator Random Variable
 The Indicator Function

4 The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

A "Random Process"

First

In order to exemplify the usage of the probabilistic analysis, we will use the "Hiring Problem."

Why

It is clear that hiring a person is a random process.

An Example

Many possible process involving a "Expected" count in the number of steps of them.



< ロ > < 回 > < 回 > < 回 > < 回 >

A "Random Process"

First

In order to exemplify the usage of the probabilistic analysis, we will use the "Hiring Problem."

Why?

It is clear that hiring a person is a random process.

An Example

Many possible process involving a "Expected" count in the number of steps of them.



イロト イヨト イヨト

A "Random Process"

First

In order to exemplify the usage of the probabilistic analysis, we will use the "Hiring Problem."

Why?

It is clear that hiring a person is a random process.

An Example

Many possible process involving a "Expected" count in the number of steps of them.



Suppose the following

• You are using an employment agency to hire a new office assistant.

not to hire that person.



Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

・ロン ・四 と ・ 回 と ・

5/27

Cost to interview is c_i per candidate

- Cost to hire is c_h per candidate.
- Assume that $c_h > c_i.$

Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

Cost

• Cost to interview is c_i per candidate.

You must have somebody working all the time!

You will always hire the best candidate that you interview!

イロン 不通 とうせい イロン

Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

Cost

- Cost to interview is c_i per candidate.
- Cost to hire is c_h per candidate.

You must have somebody working all the time!

You will always hire the best candidate that you interview!

Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

Cost

- Cost to interview is c_i per candidate.
- Cost to hire is c_h per candidate.
- Assume that $c_h > c_i$.

You must have somebody working all the time!

You will always hire the best candidate that you interview!

Cinvestar かくの 5 / 27

Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

Cost

- Cost to interview is c_i per candidate.
- Cost to hire is c_h per candidate.
- Assume that $c_h > c_i$.

Observation!

You must have somebody working all the time!

You will always hire the best candidate that you interview!

Suppose the following

- You are using an employment agency to hire a new office assistant.
- You interview the candidate and must immediately decide whether or not to hire that person.

Cost

- Cost to interview is c_i per candidate.
- Cost to hire is c_h per candidate.
- Assume that $c_h > c_i$.

Observation!

- You must have somebody working all the time!
- You will always hire the best candidate that you interview!

A D > A D > A D > A D >

Outline

- The Hiring Problem
 Introduction
- 2 The Hiring Algorithm• An Initial Algorithm
 - What do we want?
- Indicator Random Variable
 The Indicator Function
- 4 The Randomized Hiring
 - Introduction
 - The Basic Algorithm
- 5 Methods to Enforce Randomization
 - Permute By Sorting
 - Permute in Place



< ロ > < 同 > < 回 > < 回 >

Hire-Assistant(n)

```
    best = 0 ▷ candidate dummy
    for i == 1 to iii
    interview iii
    if candidate i is better than candidate b
    best == i
    bire candidate i
```



Hire-Assistant(n)

- $best = 0 \triangleright candidate dummy$
- $\label{eq:integral} \textbf{0} \ \text{ for } i=1 \ \text{to} \ n$

if candidate i is better than candidate best

- best = i
 - hire candidate i

Given n candidates and we hire m of them,

 $O\left(nc_i + mc_h\right)$



Hire-Assistant(n)

- $best = 0 \triangleright candidate dummy$
- (2) for i = 1 to n
- interview i
 - if candidate i is better than candidate best
 - best = i
 - hire candidate i

Given n candidates and we hire m of them,

 $O\left(nc_i + mc_h\right)$



Hire-Assistant(n)

- $best = 0 \triangleright candidate dummy$
- (2) for i = 1 to n
- interview i
- \bullet if candidate *i* is better than candidate best

Given $m{n}$ candidates and we hire $m{m}$ of them,

 $O\left(nc_i+mc_h\right)$



Hire-Assistant(n)

- $best = 0 \triangleright candidate dummy$
- (a) for i = 1 to n

6

- interview i
- \bullet if candidate *i* is better than candidate best

$$best = i$$

nire candidate (

Given $m{n}$ candidates and we hire $m{m}$ of them,

 $O\left(nc_i + mc_h\right)$



Hire-Assistant(n)

• $best = 0 \triangleright candidate d$	lummy
---	-------

- (2) for i = 1 to n
- interview i
- \bullet if candidate *i* is better than candidate best

```
\bullet best = i
```

```
\bullet \qquad \qquad \text{hire candidate } i
```

Given n candidates and we hire m of them,

 $O\left(nc_i + mc_h\right)$



Hire-Assistant(n)

1	best =	0 ⊳	candidate	dummy
---	--------	-----	-----------	-------

- 2 for i = 1 to n
- interview i
- \bullet if candidate *i* is better than candidate best

イロト イボト イヨト イヨト

3

7 / 27

- \bullet best = i
 - hire candidate i

Cost

6

Given \boldsymbol{n} candidates and we hire \boldsymbol{m} of them,

Hire-Assistant(n)

|--|--|

- (a) for i = 1 to n
- interview i
- if candidate i is better than candidate best

$$\bullet$$
 best = t

hire candidate i

Cost

6

Given \boldsymbol{n} candidates and we hire \boldsymbol{m} of them,

$$O\left(nc_i + mc_h\right)$$

3

イロト イヨト イヨト

(1)

Outline

The Hiring ProblemIntroduction

- 2 The Hiring AlgorithmAn Initial Algorithm
 - What do we want?
- Indicator Random Variable
 The Indicator Function

4 The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

We have that

- We need to find the maximum or minimum in a sequence by examining each element and maintaining a current "WINNER."
 - which element is currently winning.



We have that

- We need to find the maximum or minimum in a sequence by examining each element and maintaining a current "WINNER."
- The variable *m* denotes how many times we change our notion of which element is currently winning.

Worst Case Analysis

• You hire all of n

$$O(nc_i + nc_h) = O(nc_h)$$

Why? Because every time we hire somebody, we fire somebody.



We have that

- We need to find the maximum or minimum in a sequence by examining each element and maintaining a current "WINNER."
- The variable *m* denotes how many times we change our notion of which element is currently winning.

Worst Case Analysis

• You hire all of n

$O(nc_i + nc_h) = O(nc_h)$

Why? Because every time we hire somebody, we fire somebody



イロト イヨト イヨト

We have that

- We need to find the maximum or minimum in a sequence by examining each element and maintaining a current "WINNER."
- The variable *m* denotes how many times we change our notion of which element is currently winning.

Worst Case Analysis

• You hire all of n

$$O(nc_i + nc_h) = O(nc_h)$$

Why? Because every time we hire somebody, we fire somebody

イロト イヨト イヨト

We have that

- We need to find the maximum or minimum in a sequence by examining each element and maintaining a current "WINNER."
- The variable *m* denotes how many times we change our notion of which element is currently winning.

Worst Case Analysis

• You hire all of n

$$O(nc_i + nc_h) = O(nc_h)$$

• Why? Because every time we hire somebody, we fire somebody.



We want to avoid the worst case

How?

Because many times we do not get the worst case

Actually

Many times we get the "Average" input

This makes

The probability analysis a useful tool to analyze average complexities for many algorithms



We want to avoid the worst case

How?

Because many times we do not get the worst case

Actually

Many times we get the "Average" input

This makes

The probability analysis a useful tool to analyze average complexities for many algorithms



We want to avoid the worst case

How?

Because many times we do not get the worst case

Actually

Many times we get the "Average" input

This makes

The probability analysis a useful tool to analyze average complexities for many algorithms



イロト イヨト イヨト

Uniform Distribution Assumption

 Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
 The input in the hiring problem comes from a uniform distribution.

Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- **2** The possible number of permutations of individuals by the rank is n!

to have any ranking - Uniform Distribution Assumption.

Essentials of Probability Analysis

- You assume a distribution over permutation of elements.
- The expectation is over this distribution.
- This technique requires that we can make a reasonable characterization of the input distribution.



Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- **2** The possible number of permutations of individuals by the rank is n!
- Therefore, if we assume that all individuals have the same probability to have any ranking Uniform Distribution Assumption.

Essentials of Probability Analysis

- You assume a distribution over permutation of elements.
- The expectation is over this distribution.
- This technique requires that we can make a reasonable characterization of the input distribution.



Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- **2** The possible number of permutations of individuals by the rank is n!
- Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
- O The input in the hiring problem comes from a uniform distribution.

Essentials of Probability Analysis

- You assume a distribution over permutation of elements.
- The expectation is over this distribution.
- This technique requires that we can make a reasonable characterization of the input distribution.



Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- 2 The possible number of permutations of individuals by the rank is n!
- Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
- O The input in the hiring problem comes from a uniform distribution.

Essentials of Probability Analysis

• You assume a distribution over permutation of elements.

This technique requires that we can make a reasonable characterization of the input distribution.



Probabilistic Analysis

Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- 2 The possible number of permutations of individuals by the rank is n!
- Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.

O The input in the hiring problem comes from a uniform distribution.

Essentials of Probability Analysis

- You assume a distribution over permutation of elements.
- The expectation is over this distribution.

Probabilistic Analysis

Uniform Distribution Assumption

- $\textbf{O} \text{ Assign a } rank \text{ to each candidate } i, rank: U \rightarrow \{1,2,...,n\}$
- 2 The possible number of permutations of individuals by the rank is n!
- Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
- O The input in the hiring problem comes from a uniform distribution.

Essentials of Probability Analysis

- You assume a distribution over permutation of elements.
- The expectation is over this distribution.
- This technique requires that we can make a reasonable characterization of the input distribution.



Outline

The Hiring ProblemIntroduction

2 The Hiring Algorithm

- An Initial Algorithm
- What do we want?

Indicator Random Variable The Indicator Function

The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

Indicator Random Variable

• We require to use the indicator random variable to facilitate the use of probabilistic analysis:

Using this we have



イロト イヨト イヨト イヨト

Indicator Random Variable

• We require to use the indicator random variable to facilitate the use of probabilistic analysis:

$$I\left\{A\right\} = \begin{cases} 0 & \text{if A does not ocurr} \\ 1 & \text{if A does ocurr} \end{cases}$$

Using this we have

Lemma b.1

Given a sample space S and an event A in the sample space S, let $X_A = I \{A\}$. Then $E[X_A] = Pr\{A\}$.



Indicator Random Variable

• We require to use the indicator random variable to facilitate the use of probabilistic analysis:

$$I\left\{A
ight\} = egin{cases} 0 & ext{if A does not ocurr} \ 1 & ext{if A does ocurr} \end{cases}$$

• Using this we have

Lemma b.1

Given a sample space S and an event A in the sample space S, let $X_A = I \{A\}$. Then $E[X_A] = Pr\{A\}$.



Indicator Random Variable

• We require to use the indicator random variable to facilitate the use of probabilistic analysis:

$$I \{A\} = \begin{cases} 0 & \text{if A does not ocurr} \\ 1 & \text{if A does ocurr} \end{cases}$$

• Using this we have

Lemma 5.1

Given a sample space S and an event A in the sample space S, let $X_A = I \{A\}$. Then $E[X_A] = Pr\{A\}$.



A D > A D > A D > A D >

Given X

Assume a X, the random variable of the number of times we hire a person.

イロト イロト イヨト イヨト

э

14 / 27

Given X

Assume a X, the random variable of the number of times we hire a person.

$$E[X] = \sum_{x=1}^{n} x Pr\{X = x\}$$
 (2)

We could analyze the hiring problem by using the indicator function:

 $X_i = I \{ \text{ candidate i is hired} \} = egin{cases} 1 & ext{if candidate i is hired} \ 0 & ext{if candidate i is not hired} \end{cases}$



Given X

Assume a X, the random variable of the number of times we hire a person.

$$E[X] = \sum_{x=1}^{n} x Pr\{X = x\}$$
 (2)

A D > A D > A D > A D >

Then

We could analyze the hiring problem by using the indicator function:



Given X

Assume a X, the random variable of the number of times we hire a person.

$$E[X] = \sum_{x=1}^{n} x Pr\{X = x\}$$
 (2)

A D > A D > A D > A D >

Then

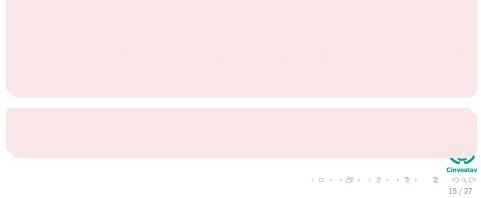
We could analyze the hiring problem by using the indicator function:

$$X_i = I \{ \text{ candidate i is hired} \} = \begin{cases} 1 & \text{ if candidate i is hired} \\ 0 & \text{ if candidate i is not hired} \end{cases}$$
(3)



Decomposing Complex Indicator Variables

$$X = X_1 + X_2 + \dots + X_n.$$



Decomposing Complex Indicator Variables

$$X = X_1 + X_2 + \dots + X_n.$$

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

 $E\left[X_{i}
ight]=Pr\left\{
ight.$ cantidate i is hired $ight\}$:



Decomposing Complex Indicator Variables

$$X = X_1 + X_2 + \dots + X_n.$$

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

$$E[X_i] = Pr \{ \text{ cantidate } i \text{ is hired} \} = \frac{1}{i}$$
(4)



Decomposing Complex Indicator Variables

$$X = X_1 + X_2 + \ldots + X_n.$$

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

$$E[X_i] = Pr\{ \text{ cantidate } i \text{ is hired}\} = \frac{1}{i}$$
 (4)

Why?

If we hire a new i, this candidate is better than the previous 1 to i - 1.

イロト 不得 トイヨト イヨト

Finally, we can calculate

$$E[X] = E\left[\sum_{i=1}^{n} X_{i}\right]$$

$$= \sum_{i=1}^{n} E[X]$$

$$= \sum_{i=1}^{n} \frac{1}{i}$$

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

$$= 1 + \ln n$$

16 / 27

Finally, we can calculate

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right]$$
$$= \sum_{i=1}^{n} E[X]$$

<ロト < 回ト < 目ト < 目ト < 目ト 目 のQで 16/27

Finally, we can calculate

$$E[X] = E\left[\sum_{i=1}^{n} X_{i}\right]$$
$$= \sum_{i=1}^{n} E[X]$$
$$= \sum_{i=1}^{n} \frac{1}{i}$$

Finally, we can calculate

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right]$$
$$= \sum_{i=1}^{n} E[X]$$
$$= \sum_{i=1}^{n} \frac{1}{i}$$
$$= 1 + \sum_{i=2}^{n} \frac{1}{i}$$

 $= 1 - \ln n$

E

Finally, we can calculate

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right]$$
$$= \sum_{i=1}^{n} E[X]$$
$$= \sum_{i=1}^{n} \frac{1}{i}$$
$$= 1 + \sum_{i=2}^{n} \frac{1}{i}$$
$$\leq 1 + \int_{1}^{n} \frac{1}{i} di$$

<ロト < 団ト < 巨ト < 巨ト < 巨ト 三 のへで 16/27

E

Finally, we can calculate

$$[X] = E\left[\sum_{i=1}^{n} X_{i}\right]$$
$$= \sum_{i=1}^{n} E[X]$$
$$= \sum_{i=1}^{n} \frac{1}{i}$$
$$= 1 + \sum_{i=2}^{n} \frac{1}{i}$$
$$\leq 1 + \int_{1}^{n} \frac{1}{i} di$$
$$= 1 + \ln n$$

< □ ▶ < □ ▶ < ■ ▶ < ■ ▶ < ■ ▶ 16/27



We have then

$$E[X] = \ln n + O(1)$$

(5)

Thus

Final hiring cost is $O(c_h \ln n)$.





We have then

$$E[X] = \ln n + O(1)$$

Thus

Final hiring cost is $O(c_h \ln n)$.



(5)

Outline

The Hiring ProblemIntroduction

2 The Hiring Algorithm

- An Initial Algorithm
- What do we want?
- Indicator Random Variable
 The Indicator Function

The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

What if

 $\hfill \bullet$ The employment agency sends us a list of all n candidates in advance.

order, we take control of the process and enforce a random order.

What if

- $\bullet\,$ The employment agency sends us a list of all n candidates in advance.
- On each day, we randomly choose a candidate from the list to interview.
 - Instead of relaying on the candidate being presented to us in a random order, we take control of the process and enforce a random order.

What makes an algorithm randomized?

- An algorithm is randomized if its behavior is determined in part by values produced by a random-number generator.
- A random-number generator is implemented by a pseudorandom-number generator, which is a deterministic method returning numbers that "look" random and can pass certain statistical tests.

What if

- The employment agency sends us a list of all n candidates in advance.
- On each day, we randomly choose a candidate from the list to interview.
- Instead of relaying on the candidate being presented to us in a random order, we take control of the process and enforce a random order.

What makes an algorithm randomized

- An algorithm is randomized if its behavior is determined in part by values produced by a random-number generator.
- A random-number generator is implemented by a pseudorandom-number generator, which is a deterministic method returning numbers that "look" random and can pass certain statistical tests.

What if

- The employment agency sends us a list of all n candidates in advance.
- On each day, we randomly choose a candidate from the list to interview.
- Instead of relaying on the candidate being presented to us in a random order, we take control of the process and enforce a random order.

What makes an algorithm randomized?

• An algorithm is **randomized** if its behavior is determined in part by values produced by a random-number generator.

oseudorandom-number generator, which is a deterministic method returning numbers that "look" random and can pass certain statistic rests.

What if

- The employment agency sends us a list of all n candidates in advance.
- On each day, we randomly choose a candidate from the list to interview.
- Instead of relaying on the candidate being presented to us in a random order, we take control of the process and enforce a random order.

What makes an algorithm randomized?

- An algorithm is **randomized** if its behavior is determined in part by values produced by a random-number generator.
- A random-number generator is implemented by a pseudorandom-number generator, which is a deterministic method returning numbers that "look" random and can pass certain statistical tests.

Outline

The Hiring ProblemIntroduction

2 The Hiring Algorithm

- An Initial Algorithm
- What do we want?
- Indicator Random VariableThe Indicator Function

The Randomized Hiring

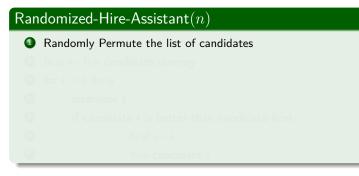
- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

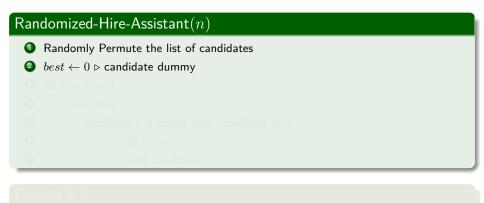
- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

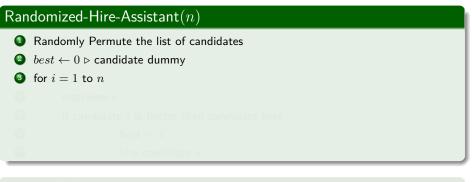






The expected hiring cost of the procedure Randomized-Hiring-Assistant is

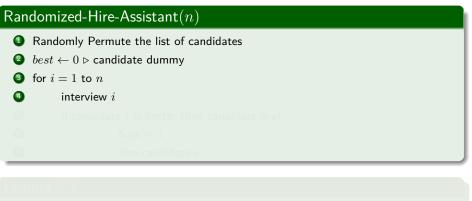
 $O(c_h \ln n)$.



Lemma 5.3

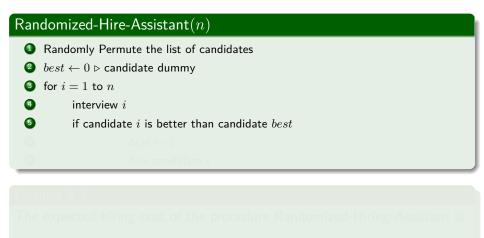
The expected hiring cost of the procedure Randomized-Hiring-Assistant is

 $O\left(c_{h}\ln n\right)$.

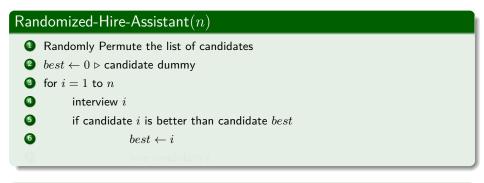


The expected hiring cost of the procedure Randomized-Hiring-Assistant is

 $O(c_h \ln n)$.



 $O(c_h \ln n)$.



he expected hiring cost of the procedure Randomized-Hiring-A

 $O\left(c_{h}\ln n\right)$.

The Random Hiring Algorithm

Randomized-Hire-Assistant (n)		
Randomly Per	mute the list of candidates	
2 $best \leftarrow 0 \triangleright candidate dummy$		
(a) for $i = 1$ to n		
Interview	r i	
if candid	ate i is better than candidate $best$	
6	$best \leftarrow i$	
0	hire candidate <i>i</i>	

Lemma 5.3

The expected hiring cost of the procedure Randomized-Hiring-Assistant is

 $O(c_h \ln n)$.

Cinvestav

21 / 27

2

イロト イロト イヨト イヨト

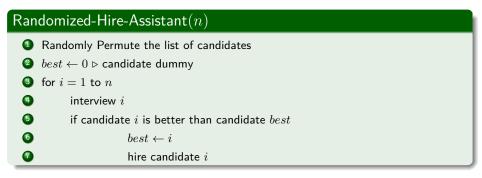
The Random Hiring Algorithm

$Randomized\operatorname{-Hire-Assistant}(n)$		
1	Andomly Permute the list of candidates	
2 $best \leftarrow 0 \triangleright candidate dummy$		
If for $i = 1$ to n		
4	interview i	
6	if candidate i is better than candidate $best$	
6	$best \leftarrow i$	
0	hire candidate <i>i</i>	

Lemma 5.3

The expected hiring cost of the procedure Randomized-Hiring-Assistant is

The Random Hiring Algorithm



Lemma 5.3

The expected hiring cost of the procedure Randomized-Hiring-Assistant is

 $O(c_h \ln n)$.

(6)

イロト イヨト イヨト

Outline

The Hiring ProblemIntroduction

- 2 The Hiring Algorithm
 - An Initial Algorithm
 - What do we want?
- Indicator Random Variable
 The Indicator Function

4 The Randomized Hiring

- Introduction
- The Basic Algorithm

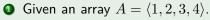
5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

Process



 $lacksymbol{eta}$ Generate a random ranking P

 $B = \langle 2, 4, 1, 3 \rangle$

Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.

) Then, sort A using the P ranking

 $B = \langle 2, 4, 1, 3 \rangle$

Permute-By-Sorting(A

- n = lenght[A]
- \bigcirc for i = 1 to n
- \bigcirc do $P[i] = RANDOM(1, n^3)$
- \bigcirc sort A, using P as sort keys
- \bigcirc return A

Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.
- O Then, sort A using the P ranking

$$B = \langle 2, 4, 1, 3 \rangle$$

Permute-By-Sorting(A

- $\bigcirc n = lenght[A]$
- \bigcirc for i = 1 to n
- \bigcirc do $P[i] = RANDOM(1, n^3)$
- \bigcirc sort A, using P as sort keys
- \bigcirc return A

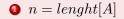
(7)

Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.

$$B = \langle 2, 4, 1, 3 \rangle$$

Permute-By-Sorting(A)



do P[i] = RANDOM(1, n)

) sort A, using P as sort keys

return A

(7)

din year di

Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.

$$B = \langle 2, 4, 1, 3 \rangle$$

イロト 不得 トイヨト イヨト

(7)

III yeard

-

Permute-By-Sorting(A)

n = lenght[A]
for i = 1 to n

Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.
- O Then, sort A using the P ranking

$$B = \langle 2, 4, 1, 3 \rangle \tag{7}$$

Permute-By-Sorting(A)

э

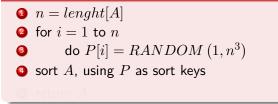
Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.

$$B = \langle 2, 4, 1, 3 \rangle \tag{7}$$

イロト イボト イヨト イヨト

Permute-By-Sorting(A)



Process

- Given an array $A = \langle 1, 2, 3, 4 \rangle$.
- **2** Generate a random ranking P.

$$B = \langle 2, 4, 1, 3 \rangle \tag{7}$$

イロト イボト イヨト イヨト

$\mathsf{Permute-By-Sorting}(A)$

n = lenght[A]
for i = 1 to n
do P[i] = RANDOM (1, n³)
sort A, using P as sort keys
return A

Proving Correctness of Permute-By-Sorting

Lemma 5.4

Procedure Permute-By-Sorting produces a uniform random permutation of the input, assuming that all priorities are distinct.



Outline

The Hiring ProblemIntroduction

- 2 The Hiring Algorithm
 - An Initial Algorithm
 - What do we want?
- Indicator Random Variable
 The Indicator Function

4 The Randomized Hiring

- Introduction
- The Basic Algorithm

5 Methods to Enforce Randomization

- Permute By Sorting
- Permute in Place



< ロ > < 同 > < 回 > < 回 >

Algorithm

Randomize-In-Place(A)

- $\bullet \ n = lenght[A]$
- ${\rm 2 \hspace{-0.5mm} for} \ i=1 \ {\rm to} \ n$

Lemma 5.5

Procedure Randomize-In-Place computes a uniform random permutation.



Algorithm

Randomize-In-Place(A)

- $\bullet \ n = lenght[A]$
- ${\rm 2 \hspace{-0.5mm} of} \ {\rm for} \ i=1 \ {\rm to} \ n$

Lemma 5.5

Procedure Randomize-In-Place computes a uniform random permutation.



Exercises

- 5.1-1
- 5.1-2
- 5.2-1
- 5.2-3
- 5.2-5
- 5.3-1
- 5.3-3
- 5.3-4

