Analysis of Algorithms

Probabilistic Analysis

Andres Mendez-Vazquez

January 10, 2018

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

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.

Cinvestav

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.

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.

The Hiring Problem

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

- You will always hire the best candidate that you interview!

Cinvestav

The Hiring Problem

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 will always hire the best candidate that you interview!

The Hiring Problem

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!

The Hiring Problem

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!

The Hiring Problem

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!

The Hiring Problem

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!

The Hiring Problem

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!

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Hiring Algorithm

Hire-Assistant(n)

1. $best = 0$ ▷ candidate dummy
2. for $i = 1$ to n
3. interview i
4. if candidate i is better than candidate best
5. $best = i$
6. hire candidate i

Cinvestav

Hiring Algorithm

Hire-Assistant(n)

1. $best = 0$ ▷ candidate dummy
2. for $i = 1$ to n
3. interview i
4. if candidate i is better than candidate best
5. $best = i$
6. hire candidate i

Cost

Given n candidates and we hire m of them,

$$O(nc_i + mc_h) \tag{1}$$

Hiring Algorithm

Hire-Assistant(n)

$best = 0$ ▷ candidate dummy
for $i = 1$ to n
    interview i
    if candidate i is better than candidate best
        $best = i$
        hire candidate i

Cost

Given n candidates and we hire m of them,

$$O(nc_i + mc_h) \quad (1)$$

Hiring Algorithm

Hire-Assistant(n)

```
best = 0 ▷ candidate dummy
for i = 1 to n
        interview i
        if candidate i is better than candidate best
                best = i
                hire candidate i
```

Cost

Given n candidates and we hire m of them,

$$O\left(nc_i + mc_h\right) \quad (1)$$

Cinvestav

Hiring Algorithm

Hire-Assistant(n)

1. $best = 0$ ▷ candidate dummy
2. for $i = 1$ to n
3. interview i
4. if candidate i is better than candidate best
5. $best = i$
6. hire candidate i

Cost

Given n candidates and we hire m of them,

$$O\left(nc_i + mc_h\right) \quad (1)$$

Hiring Algorithm

Hire-Assistant(n)

```
best = 0 ▷ candidate dummy
for i = 1 to n
    interview i
    if candidate i is better than candidate best
        best = i
        hire candidate i
```

Cost

Given n candidates and we hire m of them,

$$O\left(nc_i + mc_h\right) \quad (1)$$

Hiring Algorithm

Hire-Assistant(n)

```
best = 0 ▷ candidate dummy
for i = 1 to n
    interview i
    if candidate i is better than candidate best
        best = i
        hire candidate i
```

Cost

Given n candidates and we hire m of them,

$$O\left(nc_i + mc_h\right) \tag{1}$$

Hiring Algorithm

Hire-Assistant(n)

1. $best = 0$ ▷ candidate dummy
2. for $i = 1$ to n
3. interview i
4. if candidate i is better than candidate best
5. $best = i$
6. hire candidate i

Cost

Given n candidates and we hire m of them,

$$O(nc_i + mc_h) \tag{1}$$

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

Paradigm

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.

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

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

Cinvestav

Paradigm

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.

Cinvestav

Paradigm

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.

Paradigm

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.

Paradigm

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

Cinvestav

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

Probabilistic Analysis

Uniform Distribution Assumption

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

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

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a $rank$ to each candidate i, $rank : U \rightarrow \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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.

Cinvestav

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a *rank* to each candidate i, $rank : U \rightarrow \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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.

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a $rank$ to each candidate i, $rank : U \rightarrow \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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.

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a $rank$ to each candidate i, $rank : U \rightarrow \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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.

Cinvestav

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a $rank$ to each candidate i, $rank : U \rightarrow \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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.

Probabilistic Analysis

Uniform Distribution Assumption

1. Assign a $rank$ to each candidate i, $rank : U \to \{1, 2, ..., n\}$
2. The possible number of permutations of individuals by the rank is $n!$
3. Therefore, if we assume that all individuals have the same probability to have any ranking - Uniform Distribution Assumption.
4. 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

1 The Hiring Problem
- Introduction

2 The Hiring Algorithm
- An Initial Algorithm
- What do we want?

3 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

Cinvestav

Indicator Random Variable

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

Indicator Random Variable

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 5.1

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

Indicator Random Variable

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 5.1

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

Indicator Random Variable

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 $E[X_A] = Pr\{A\}$.

Cinvestav

Analyzing Hiring By Indicator Variable

Given X

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

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

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

Cinvestav

Analyzing Hiring By Indicator Variable

Given X

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

$$E\left[X\right] = \sum_{x=1}^{n} xPr\left\{X = x\right\} \tag{2}$$

Then

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

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

Analyzing Hiring By Indicator Variable

Given X

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

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

Then

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

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

Analyzing Hiring By Indicator Variable

Given X

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

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

Then

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

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

Cinvestav

Representing Complex Indicator Variables

Decomposing Complex Indicator Variables

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

Representing Complex Indicator Variables

Decomposing Complex Indicator Variables

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

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

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

Representing Complex Indicator Variables

Decomposing Complex Indicator Variables

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

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

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

Why?

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

Cinvestav

Representing Complex Indicator Variables

Decomposing Complex Indicator Variables

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

Uniform Assumption

And because the candidates come randomly (Uniform Assumption):

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

Why?

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

Cinvestav

Analyzing Hiring By Indicator Variable

Finally, we can calculate

$$
\begin{aligned}
E[X] &= E\left[\sum_{i=1}^{n} X_i\right] \\
&= \sum_{i=1}^{n} E[X] \\
&= \sum_{i=1}^{n} \frac{1}{i} \\
&= 1 + \sum_{i=2}^{n} \frac{1}{i} \\
&\le 1 + \int_{1}^{n} \frac{1}{i}\, di \\
&= 1 + \ln n
\end{aligned}
$$

Analyzing Hiring By Indicator Variable

Finally, we can calculate

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

Analyzing Hiring By Indicator Variable

Finally, we can calculate

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

Analyzing Hiring By Indicator Variable

Finally, we can calculate

$$
\begin{aligned}
E[X] &= E\left[\sum_{i=1}^{n} X_i\right] \\
&= \sum_{i=1}^{n} E[X] \\
&= \sum_{i=1}^{n} \frac{1}{i} \\
&= 1 + \sum_{i=2}^{n} \frac{1}{i} \\
&\le 1 + \int_{1}^{n} \frac{1}{i}\, di \\
&= 1 + \ln n
\end{aligned}
$$

Analyzing Hiring By Indicator Variable

Finally, we can calculate

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

Analyzing Hiring By Indicator Variable

Finally, we can calculate

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

Finally

We have then

$$E[X] = \ln n + O(1) \quad (5)$$

Thus

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

Finally

We have then

$$E[X] = \ln n + O(1) \tag{5}$$

Thus

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

Cinvestav

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

Assume

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.

- 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.

Assume

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.

Assume

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.

Assume

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.

Assume

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

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0 \triangleright$ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0$ ▷ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \quad (6)$$

Cinvestav

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0$ ▷ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \tag{6}$$

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0$ ▷ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \tag{6}$$

Cinvestav

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

Randomly Permute the list of candidates
$best \leftarrow 0$ ▷ candidate dummy
for $i = 1$ to n
    interview i
    if candidate i is better than candidate $best$
        $best \leftarrow i$
        hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \tag{6}$$

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0$ ▷ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \tag{6}$$

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

Randomly Permute the list of candidates
$best \leftarrow 0$ ▷ candidate dummy
for $i = 1$ to n
    interview i
    if candidate i is better than candidate $best$
        $best \leftarrow i$
        hire candidate i

Lemma 5.3

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

$$O(c_h \ln n). \tag{6}$$

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0 \triangleright$ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \tag{6}$$

Cinvestav

The Random Hiring Algorithm

Randomized-Hire-Assistant(n)

1. Randomly Permute the list of candidates
2. $best \leftarrow 0$ ▷ candidate dummy
3. for $i = 1$ to n
4. interview i
5. if candidate i is better than candidate $best$
6. $best \leftarrow i$
7. hire candidate i

Lemma 5.3

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

$$O\left(c_h \ln n\right). \quad (6)$$

Outline

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

Permute By Sorting

Process

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

$$B = \langle 2, 4, 1, 3 \rangle \quad (7)$$

Permute By Sorting

Process

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

$$B = \langle 2, 4, 1, 3 \rangle \quad (7)$$

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

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

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

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

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

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

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

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

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

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

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. return A

Permute By Sorting

Process

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

$$B = \langle 2, 4, 1, 3 \rangle \quad (7)$$

Permute-By-Sorting(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do $P[i] = RANDOM\left(1, n^3\right)$
4. sort A, using P as sort keys
5. 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

1. The Hiring Problem
   - Introduction
2. The Hiring Algorithm
   - An Initial Algorithm
   - What do we want?
3. 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

Cinvestav

Algorithm

Randomize-In-Place(A)

$n = lenght[A]$
for $i = 1$ to n
    do swap $A[i] \longleftrightarrow A[RANDOM(i, n)]$

Lemma 5.5

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

Algorithm

Randomize-In-Place(A)

1. $n = lenght[A]$
2. for $i = 1$ to n
3. do swap $A[i] \longleftrightarrow A[RANDOM(i, n)]$

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