Análise de Algoritmos

Parte destes slides são adaptações de slides

do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.

Quicksort e Select Aleatorizados

CLRS Secs 7.3, 7.4 e 9.2

Particione

Rearranja $A[p \mathinner{.\,.} r]$ de modo que $p \le q \le r$ e
$A[p \mathinner{.\,.} q{-}1] \le A[q] < A[q{+}1 \mathinner{.\,.} r]$

PARTICIONE (A, p, r)
1 $\quad x \leftarrow A[r]$ $\qquad \triangleright$ x é o “pivô”
2 $\quad i \leftarrow p{-}1$
3 $\quad$ **para** $j \leftarrow p$ **até** $r - 1$ **faça**
4 $\qquad$ **se** $A[j] \le x$
5 $\qquad\quad$ **então** $i \leftarrow i + 1$
6 $\qquad\qquad A[i] \leftrightarrow A[j]$
7 $\quad A[i{+}1] \leftrightarrow A[r]$
8 $\quad$ **devolva** $i + 1$

Invariantes: no começo de cada iteração de 3–6,
(i0) $A[p \mathinner{.\,.} i] \le x$ (i1) $A[i{+}1 \mathinner{.\,.} j{-}1] > x$ (i2) $A[d] = x$

Particione

Rearranja $A[p\,..\,r]$ de modo que $p \le q \le r$ e $A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,r]$

PARTICIONE (A, p, r)

$x \leftarrow A[r]$ $\quad \triangleright$ x é o “pivô”
$i \leftarrow p{-}1$
para $j \leftarrow p$ **até** $r - 1$ **faça**
  se $A[j] \le x$
    então $i \leftarrow i + 1$
      $A[i] \leftrightarrow A[j]$
$A[i{+}1] \leftrightarrow A[r]$
devolva $i + 1$

Consumo de tempo: $\Theta(n)$ onde $n := r - p$.

Quicksort aleatorizado

```
PARTICIONE-ALEA(A, p, r)
    i ← RANDOM(p, r)
    A[i] ↔ A[r]
    devolva PARTICIONE(A, p, r)
```

Quicksort aleatorizado

PARTICIONE-ALEA(A, p, r)

```
    i ← RANDOM (p, r)
    A[i] ↔ A[r]
    devolva PARTICIONE (A, p, r)
```

QUICKSORT-ALE (A, p, r)

```
    se p < r
        então q ← PARTICIONE-ALEA (A, p, r)
              QUICKSORT-ALE (A, p, q − 1)
              QUICKSORT-ALE (A, q + 1, r)
```

Quicksort aleatorizado

PARTICIONE-ALEA(A, p, r)
1 $i \leftarrow$ RANDOM (p, r)
2 $A[i] \leftrightarrow A[r]$
3 **devolva** PARTICIONE (A, p, r)

QUICKSORT-ALE (A, p, r)
1 **se** $p < r$
2 **então** $q \leftarrow$ PARTICIONE-ALEA (A, p, r)
3 QUICKSORT-ALE $(A, p, q - 1)$
4 QUICKSORT-ALE $(A, q + 1, r)$

Consumo esperado de tempo?

Quicksort aleatorizado

PARTICIONE-ALEA(A, p, r)
    $i \leftarrow$ RANDOM (p, r)
    $A[i] \leftrightarrow A[r]$
    devolva PARTICIONE (A, p, r)

QUICKSORT-ALE (A, p, r)
    se $p < r$
        então $q \leftarrow$ PARTICIONE-ALEA (A, p, r)
            QUICKSORT-ALE $(A, p, q-1)$
            QUICKSORT-ALE $(A, q+1, r)$

Consumo esperado de tempo?

Basta contar o número esperado de comparações na linha 4 do PARTICIONE.

Consumo esperado de tempo

Basta contar o número esperado de comparações na linha 4 do PARTICIONE.

PARTICIONE (A, p, r)

1 $\quad x \leftarrow A[r]$ $\qquad \triangleright$ x é o "pivô"

2 $\quad i \leftarrow p{-}1$

3 $\quad$ **para** $j \leftarrow p$ **até** $r - 1$ **faça**

4 $\qquad$ **se** $A[j] \leq x$

5 $\qquad\quad$ **então** $i \leftarrow i + 1$

6 $\qquad\qquad A[i] \leftrightarrow A[j]$

7 $\quad A[i{+}1] \leftrightarrow A[r]$

8 $\quad$ **devolva** $i + 1$

Consumo de tempo esperado

Suponha $A[p \mathinner{.\,.} r]$ permutação de $1 \mathinner{.\,.} n$.

X_{ab} = número de comparações entre a e b na linha 4 do PARTICIONE do QUICKSORT-ALE;

Queremos calcular

$$
\begin{aligned}
X &= \text{total de comparações "} A[j] \leq x \text{"} \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{ab}
\end{aligned}
$$

Consumo de tempo esperado

Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, b\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

Consumo de tempo esperado

Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, b\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

$$\text{Pr}\,\{X_{ab}{=}1\} \;=\; \frac{1}{b-a+1} + \frac{1}{b-a+1} \;=\; \mathrm{E}[X_{ab}]$$

Consumo de tempo esperado

Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, b\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

$$\Pr\{X_{ab}=1\} = \frac{1}{b-a+1} + \frac{1}{b-a+1} = \mathrm{E}[X_{ab}]$$

$$X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{ab}$$

$\mathrm{E}[X] =$????

Consumo de tempo esperado

$$
\begin{aligned}
\mathrm{E}[X] &= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathrm{E}[X_{ab}] \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \Pr\{X_{ab}=1\} \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \frac{2}{b-a+1} \\
&= \sum_{a=1}^{n-1} \sum_{k=1}^{n-a} \frac{2}{k+1} \\
&< \sum_{a=1}^{n-1} 2\left(\frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{n}\right) \\
&< 2n\left(\frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{n}\right) < 2n\,(1+\ln n)
\end{aligned}
$$

CLRS (A.7), p.1060

Conclusões

O consumo de tempo esperado do algoritmo QUICKSORT-ALE é $\mathrm{O}(n \lg n)$.

Do exercício 7.4-4 do CLRS temos que

O consumo de tempo esperado do algoritmo QUICKSORT-ALE é $\Theta(n \lg n)$.

Select aleatorizado

PARTICIONE-ALEA(A, p, r)
    $i \leftarrow$ RANDOM (p, r)
    $A[i] \leftrightarrow A[r]$
    devolva PARTICIONE (A, p, r)

SELECT-ALEA (A, p, r, k)
    se $p < r$ **então**
        $q \leftarrow$ PARTICIONE-ALEA (A, p, r)
        se $k = q - p + 1$
            então devolva $A[q]$
        se $k < q - p + 1$
            então SELECT-ALEA $(A, p, q - 1, k)$
            senão SELECT-ALEA $(A, q + 1, r, k - (q - p + 1))$

Select aleatorizado

PARTICIONE-ALEA(A, p, r)

$i \leftarrow$ RANDOM (p, r)

$A[i] \leftrightarrow A[r]$

devolva PARTICIONE (A, p, r)

SELECT-ALEA (A, p, r, k)

se $p < r$ **então**

    $q \leftarrow$ PARTICIONE-ALEA (A, p, r)

    se $k = q - p + 1$

        então devolva $A[q]$

    se $k < q - p + 1$

        então SELECT-ALEA $(A, p, q - 1, k)$

        senão SELECT-ALEA $(A, q + 1, r, k - (q - p + 1))$

Consumo esperado de tempo?

Consumo de tempo esperado

Suponha $A[p\,..\,r]$ permutação de $1\,..\,n$.

X_{ab} = número de comparações entre a e b na linha 4 do PARTICIONE do SELECT-ALEA.

Observe que X_{ab} não é a mesma de antes.

De novo, queremos calcular

$$\begin{aligned} X &= \text{total de comparações "}A[j] \le x\text{"} \\ &= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{ab} \end{aligned}$$

Consumo de tempo esperado

Vamos supor que $k = n$.
Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, n\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

Consumo de tempo esperado

Vamos supor que $k = n$.
Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, n\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

$$\text{Pr}\,\{X_{ab}{=}1\} \;=\; \frac{1}{n-a+1} + \frac{1}{n-a+1} \;=\; \mathrm{E}[X_{ab}]$$

Consumo de tempo esperado

Vamos supor que $k = n$.
Supondo $a < b$,

$$X_{ab} = \begin{cases} 1 & \text{se primeiro pivô em } \{a, \ldots, n\} \text{ é } a \text{ ou } b \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_{ab} valer 1?

$$\Pr\{X_{ab}{=}1\} \;=\; \frac{1}{n-a+1} + \frac{1}{n-a+1} \;=\; \mathrm{E}[X_{ab}]$$

$$X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{ab}$$

$\mathrm{E}[X] =$????

Consumo de tempo esperado

$$
\begin{aligned}
\mathrm{E}[X] &= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathrm{E}[X_{ab}] = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathsf{Pr}\{X_{ab}=1\} \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \frac{2}{n-a+1} \\
&= \sum_{a=1}^{n-1} \frac{2(n-a)}{n-a+1} \\
&< \sum_{a=1}^{n-1} 2 < 2n.
\end{aligned}
$$

Consumo de tempo esperado

$$
\begin{aligned}
\mathrm{E}[X] &= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathrm{E}[X_{ab}] = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathsf{Pr}\left\{X_{ab}{=}1\right\} \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \frac{2}{n-a+1} \\
&= \sum_{a=1}^{n-1} \frac{2(n-a)}{n-a+1} \\
&< \sum_{a=1}^{n-1} 2 < 2n.
\end{aligned}
$$

Exercício: Refaça os cálculos para um k arbitrário.

Conclusões

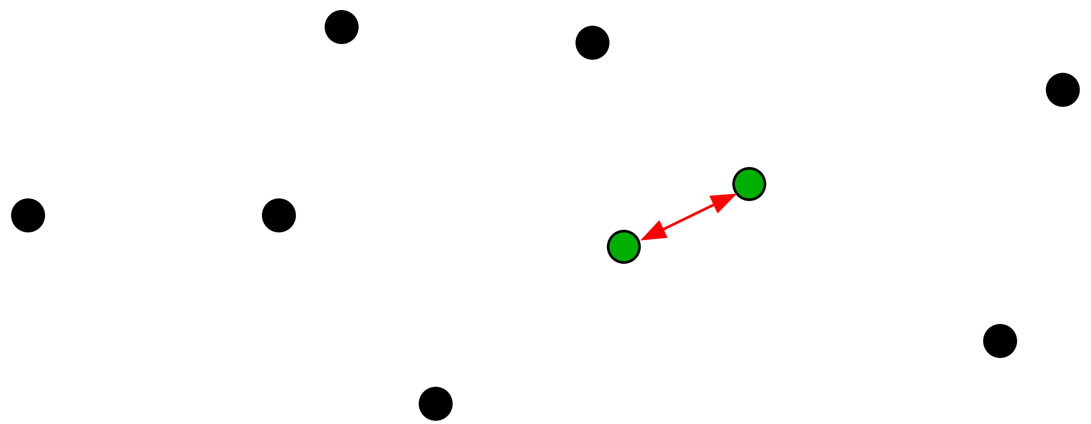
O consumo de tempo esperado do algoritmo SELECT-ALEA é $\mathrm{O}(n)$.

Par de Pontos Mais Próximos

KT Sec 13.7

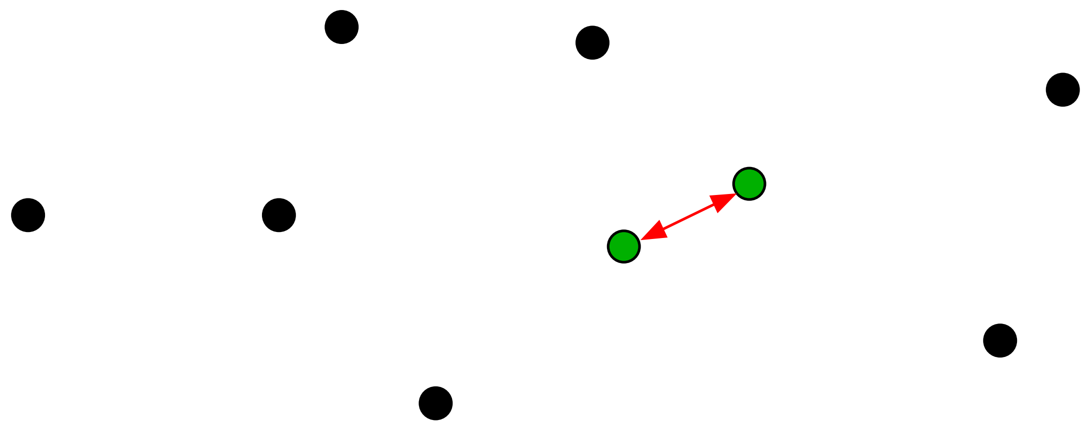
Par de pontos mais próximos

Problema: Dados n pontos no plano, determinar dois deles que estão à distância mínima.



Par de pontos mais próximos

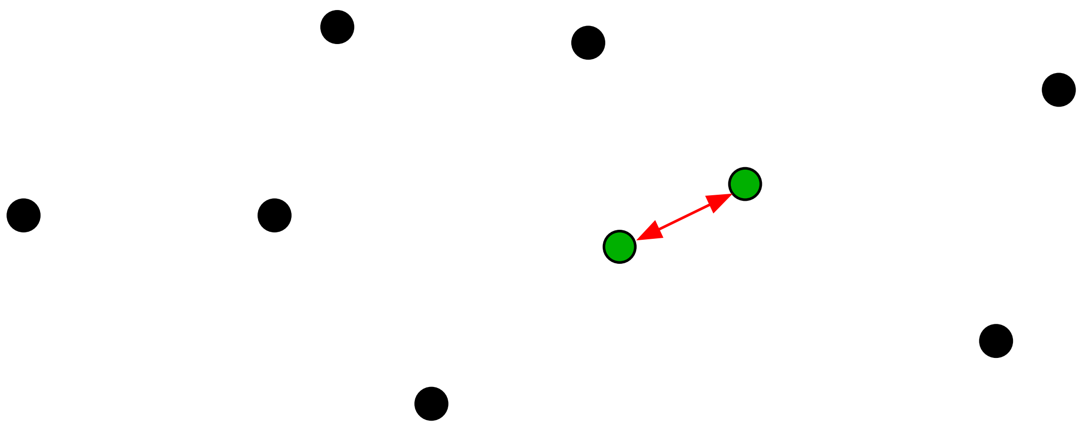
Problema: Dados n pontos no plano, determinar dois deles que estão à distância mínima.



Esta aula: algoritmo aleatorizado cujo consumo esperado de tempo é $\mathrm{O}(n)$.

Par de pontos mais próximos

Problema: Dados n pontos no plano,
determinar dois deles que estão à distância mínima.



Esta aula: algoritmo aleatorizado cujo
consumo esperado de tempo é $\mathrm{O}(n)$.

Hipótese simplificadora: todos os pontos estão no quadrado $[0,1] \times [0,1]$.

Esboço do algoritmo

Problema: Dados n pontos no quadrado $[0, 1] \times [0, 1]$, determinar dois deles que estão à distância mínima.

Esboço do algoritmo

Problema: Dados n pontos no quadrado $[0,1] \times [0,1]$, determinar dois deles que estão à distância mínima.

Seja $\delta = dist(p_1, p_2)$ e $Q_{i,k} = \left[\frac{i\delta}{2}, \frac{(i+1)\delta}{2}\right) \times \left[\frac{k\delta}{2}, \frac{(k+1)\delta}{2}\right)$, para $i, k = 0, \ldots, N$, onde $N = \lceil 2/\delta \rceil$.

Esboço do algoritmo

Problema: Dados n pontos no quadrado $[0,1] \times [0,1]$, determinar dois deles que estão à distância mínima.

Seja $\delta = dist(p_1, p_2)$ e $Q_{i,k} = \left[\frac{i\delta}{2}, \frac{(i+1)\delta}{2}\right) \times \left[\frac{k\delta}{2}, \frac{(k+1)\delta}{2}\right)$, para $i, k = 0, \ldots, N$, onde $N = \lceil 2/\delta \rceil$.

Para $j \leftarrow 3$ até n faça:
Seja S o conjunto de pares (i, k) tais que existe um ponto p_ℓ com $\ell < j$ no quadrado $Q_{i,k}$.

Esboço do algoritmo

Problema: Dados n pontos no quadrado $[0,1] \times [0,1]$, determinar dois deles que estão à distância mínima.

Seja $\delta = dist(p_1, p_2)$ e $Q_{i,k} = \left[\frac{i\delta}{2}, \frac{(i+1)\delta}{2}\right) \times \left[\frac{k\delta}{2}, \frac{(k+1)\delta}{2}\right)$, para $i, k = 0, \ldots, N$, onde $N = \lceil 2/\delta \rceil$.

Para $j \leftarrow 3$ até n faça:

Seja S o conjunto de pares (i, k) tais que existe um ponto p_ℓ com $\ell < j$ no quadrado $Q_{i,k}$.

Calcule (r, s) tal que p_j está em $Q_{r,s}$.

Para cada (i, k) em S tal que $|r - i| \leq 2$ e $|s - k| \leq 2$
calcule $dist(p_j, p_\ell)$ onde $\ell < j$ e $p_\ell \in Q_{i,k}$
e atualize δ quando necessário.

Esboço do algoritmo

Problema: Dados n pontos no quadrado $[0,1] \times [0,1]$, determinar dois deles que estão à distância mínima.

Seja $\delta = dist(p_1, p_2)$ e $Q_{i,k} = \left[\frac{i\delta}{2}, \frac{(i+1)\delta}{2}\right) \times \left[\frac{k\delta}{2}, \frac{(k+1)\delta}{2}\right)$, para $i, k = 0, \ldots, N$, onde $N = \lceil 2/\delta \rceil$.

Para $j \leftarrow 3$ até n faça:
  Seja S o conjunto de pares (i, k) tais que existe um ponto p_ℓ com $\ell < j$ no quadrado $Q_{i,k}$.

  Calcule (r, s) tal que p_j está em $Q_{r,s}$.

  Para cada (i, k) em S tal que $|r - i| \leq 2$ e $|s - k| \leq 2$
    calcule $dist(p_j, p_\ell)$ onde $\ell < j$ e $p_\ell \in Q_{i,k}$
    e atualize δ quando necessário.

Devolva δ.

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Que operações sofre S?

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Que operações sofre S?

Por iteração, uma inserção e algumas consultas.

Em algumas iterações, S muda totalmente...

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Que operações sofre S?

Por iteração, uma inserção e algumas consultas.

Em algumas iterações, S muda totalmente...

Seria bom se...
inserções e buscas consumissem tempo (esperado) $\mathrm{O}(1)$.

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Que operações sofre S?

Por iteração, uma inserção e algumas consultas.

Em algumas iterações, S muda totalmente...

Seria bom se...
inserções e buscas consumissem tempo (esperado) $\mathrm{O}(1)$.

Que ED atinge isso?

Consumo de tempo estimado

Como fazer para o consumo esperado de tempo ser $\mathrm{O}(n)$?

Que ED usar para armazenar S?

Que operações sofre S?

Por iteração, uma inserção e algumas consultas.

Em algumas iterações, S muda totalmente...

Seria bom se...
inserções e buscas consumissem tempo (esperado) $\mathrm{O}(1)$.

Que ED atinge isso? Hashing!

Primeira tentativa

DISTÂNCIA(p, n)
1 $\delta \leftarrow dist(p_1, p_2)$
2 seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$
3 CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)

Primeira tentativa

DISTÂNCIA(p, n)

$\delta \leftarrow dist(p_1, p_2)$

seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$

CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)

para $j \leftarrow 3$ **até** n **faça**

    seja (r, s) tal que $p_j \in Q_{r,s}$

    para $t \leftarrow -2$ **até** 2 **faça**

        para $u \leftarrow -2$ **até** 2 **faça**

            se PERTENCE$(H, r + t, s + u)$

Primeira tentativa

DISTÂNCIA(p, n)

$\delta \leftarrow dist(p_1, p_2)$
seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$
CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)
para $j \leftarrow 3$ **até** n **faça**
    seja (r, s) tal que $p_j \in Q_{r,s}$
    para t, $u \leftarrow -2$ **até** 2 **faça**
        se PERTENCE$(H, r + t, s + u)$

Primeira tentativa

DISTÂNCIA(p, n)

$\delta \leftarrow dist(p_1, p_2)$

seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$

CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)

para $j \leftarrow 3$ **até** n **faça**

  seja (r, s) tal que $p_j \in Q_{r,s}$

  para t, $u \leftarrow -2$ **até** 2 **faça**

    se PERTENCE$(H, r + t, s + u)$

      então seja $p_\ell \in Q_{r+t, s+u}$ com $\ell < j$

        se $\delta > dist(p_j, p_\ell)$ **então** $\delta \leftarrow dist(p_j, p_\ell)$

Primeira tentativa

DISTÂNCIA(p, n)

$\delta \leftarrow dist(p_1, p_2)$
seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$
CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)
para $j \leftarrow 3$ **até** n **faça**
    seja (r, s) tal que $p_j \in Q_{r,s}$
    para t, $u \leftarrow -2$ **até** 2 **faça**
        se PERTENCE$(H, r+t, s+u)$
            então seja $p_\ell \in Q_{r+t, s+u}$ com $\ell < j$
                se $\delta > dist(p_j, p_\ell)$ **então** $\delta \leftarrow dist(p_j, p_\ell)$
    se δ foi alterado nessa iteração
        então RECONSTRUA(H, p, j)
        senão INSIRA(H, r, s, p_j)
devolva δ

Primeira tentativa

DISTÂNCIA(p, n)

1 $\delta \leftarrow dist(p_1, p_2)$
2 seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$
3 CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)
4 **para** $j \leftarrow 3$ **até** n **faça**
5   seja (r, s) tal que $p_j \in Q_{r,s}$
6   **para** t, $u \leftarrow -2$ **até** 2 **faça**
7   **se** PERTENCE$(H, r+t, s+u)$
8   **então** seja $p_\ell \in Q_{r+t,s+u}$ com $\ell < j$
9   **se** $\delta > dist(p_j, p_\ell)$ **então** $\delta \leftarrow dist(p_j, p_\ell)$
10   **se** δ foi alterado nessa iteração
11   **então** RECONSTRUA(H, p, j)
12   **senão** INSIRA(H, r, s, p_j)
13 **devolva** δ

Consumo de tempo esperado: $O(n)$ exceto pela linha 11.

Versão final

DISTÂNCIA(p, n)

EMBARALHE(p, n) ▷ permutação aleatória dos pontos dados

$\delta \leftarrow dist(p_1, p_2)$

seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$

CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)

para $j \leftarrow 3$ **até** n **faça**

    seja (r, s) tal que $p_j \in Q_{r,s}$

    para t, $u \leftarrow -1$ **até** 1 **faça**

        se PERTENCE$(H, r + t, s + u)$

            então seja $p_\ell \in Q_{r+t, s+u}$ com $\ell < j$

                se $\delta < dist(p_j, p_\ell)$ **então** $\delta \leftarrow dist(p_j, p_\ell)$

    se δ foi alterado nessa iteração

        então RECONSTRUA(H, p, j)

        senão INSIRA(H, r, s, p_j)

devolva δ

Versão final

DISTÂNCIA(p, n)

EMBARALHE(p, n) $\triangleright$ permutação aleatória dos pontos dados
$\delta \leftarrow dist(p_1, p_2)$
seja (i_ℓ, k_ℓ) tal que $p_\ell \in Q_{i_\ell, k_\ell}$ para $\ell = 1, 2$
CRIE-HASHING(H) INSIRA(H, i_1, k_1, p_1) INSIRA(H, i_2, k_2, p_2)
para $j \leftarrow 3$ **até** n **faça**
    seja (r, s) tal que $p_j \in Q_{r,s}$
    para t, $u \leftarrow -1$ **até** 1 **faça**
        se PERTENCE$(H, r + t, s + u)$
            então seja $p_\ell \in Q_{r+t, s+u}$ com $\ell < j$
                se $\delta < dist(p_j, p_\ell)$ **então** $\delta \leftarrow dist(p_j, p_\ell)$
    se δ foi alterado nessa iteração
        então RECONSTRUA(H, p, j)
        senão INSIRA(H, r, s, p_j)
devolva δ

Qual é o consumo de tempo esperado?

Consumo de tempo

Seja X o número de inserções em H.

O consumo de tempo é proporcional a $\mathrm{E}[X]$.

Consumo de tempo

Seja X o número de inserções em H.

O consumo de tempo é proporcional a $\mathrm{E}[X]$.

Seja X_j a variável binária que vale 1 sse
a linha 12 foi executada na iteração j.

$$X = \sum_{j=1}^{n} (1 + (j-1)X_j) \leq n + \sum_{j=1}^{n} j\, X_j$$

Consumo de tempo

Seja X o número de inserções em H.

O consumo de tempo é proporcional a $\mathrm{E}[X]$.

Seja X_j a variável binária que vale 1 sse
a linha 12 foi executada na iteração j.

$$X \;=\; \sum_{j=1}^{n} (1 + (j-1)X_j) \;\leq\; n + \sum_{j=1}^{n} j\, X_j$$

Mas então

$$\mathrm{E}[X] \;\leq\; n + \sum_{j=1}^{n} j\, \mathrm{E}[X_j] \;=\; n + \sum_{j=1}^{n} j\, \Pr\{X_j = 1\}.$$

Consumo de tempo

Seja X o número de inserções em H.

O consumo de tempo é proporcional a $\mathrm{E}[X]$.

Seja X_j a variável binária que vale 1 sse
a linha 12 foi executada na iteração j.

$$X = \sum_{j=1}^{n}(1+(j-1)X_j) \le n + \sum_{j=1}^{n} j\,X_j$$

Mas então

$$\mathrm{E}[X] \le n + \sum_{j=1}^{n} j\,\mathrm{E}[X_j] = n + \sum_{j=1}^{n} j\,\Pr\{X_j = 1\}.$$

Note que $\Pr\{X_j = 1\} \le 2/j$, logo $\mathrm{E}[X] \le n + 2n = 3n$.