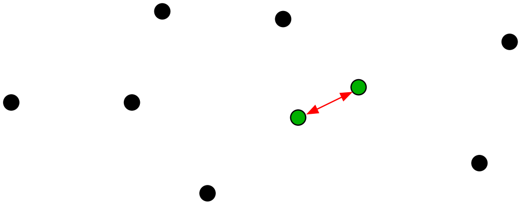
Aula 16

Par de Pontos Mais Próximos

Sec 13.7 do livro de Kleinberg e Tardos

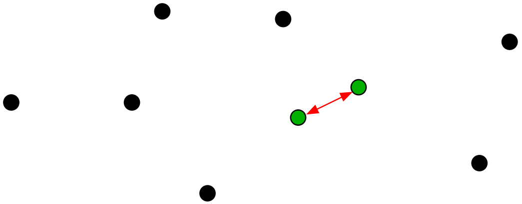
Par de pontos mais próximos

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



Par de pontos mais próximos

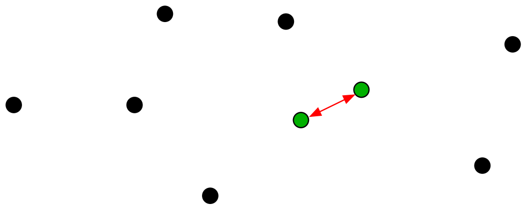
Problema: Dados n pontos no plano,
determinar dois deles que estão à distância mínima.



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

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 $O(n)$.

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 $Q_{i,k} = (\frac{i\delta}{2}, \frac{(i+1)\delta}{2}) \times (\frac{k\delta}{2}, \frac{(k+1)\delta}{2})$, 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 $\delta = \mathit{dist}(p_1, p_2)$ 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 $\mathit{dist}(p_j, p_\ell)$ 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)

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

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)

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)

$\delta \leftarrow \mathit{dist}(p_1, p_2)$
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 $\delta \leftarrow \mathit{dist}(p_1, p_2)$
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** $\delta > \mathit{dist}(p_j, p_\ell)$ **então** $\delta \leftarrow \mathit{dist}(p_j, p_\ell)$
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: $\mathrm{O}(n)$ exceto pela linha 11.

Versão final

Distância(p, n)

Embaralhe(p, n) $\triangleright$ permutação aleatória dos pontos dados
$\delta \leftarrow \mathit{dist}(p_1, p_2)$
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 $\delta < \mathit{dist}(p_j, p_\ell)$ **então** $\delta \leftarrow \mathit{dist}(p_j, p_\ell)$
    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) ▷ 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.

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

Mas então

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

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