Tópicos de Análise de Algoritmos

AULA 11

Matroides e algoritmos de clustering

Sec 4.5 e 4.7 do KT e CLRS 16.4 do CLRS

Algoritmo de Kruskal

Problema: Dado um grafo $G = (V, E)$ conexo,
e um custo $c_e > 0$ para cada aresta e em E,
encontrar uma árvore geradora mínima.

Árvore geradora mínima: árvore geradora de custo mínimo.

Algoritmo de Kruskal

Problema: Dado um grafo $G = (V, E)$ conexo,
e um custo $c_e > 0$ para cada aresta e em E,
encontrar uma árvore geradora mínima.

Árvore geradora mínima: árvore geradora de custo mínimo.

```
MST-KRUSKAL (V, E, c)
1   e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2   T ← ∅        i ← 0
3   enquanto G' = (V, T) não é conexo faça
4         i ← i + 1
5         se T ∪ {e_i} não tem circuito
6               então T ← T ∪ {e_i}
7   devolva T
```

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Problema 1: Dados U, $\mathcal{C}$, e um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Problema 1: Dados U, $\mathcal{C}$, e um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$

1 $U_1 \leftarrow \{e \in U : w_e = 1\}$ $S \leftarrow \emptyset$

2 **enquanto** existe e em U_1 tal que $S \cup \{e\} \in \mathcal{C}$ **faça**

3 $S \leftarrow S \cup \{e\}$

4 **devolva** S

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Problema 1: Dados U, $\mathcal{C}$, e um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$

1 $U_1 \leftarrow \{e \in U : w_e = 1\}$ $S \leftarrow \emptyset$

2 **enquanto** existe e em U_1 tal que $S \cup \{e\} \in \mathcal{C}$ **faça**

3 $S \leftarrow S \cup \{e\}$

4 **devolva** S

GULOSO encontra um conjunto de peso maximal de $\mathcal{C}$.

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Problema 1: Dados U, $\mathcal{C}$, e um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$

1 $U_1 \leftarrow \{e \in U : w_e = 1\}$ $S \leftarrow \emptyset$

2 **enquanto** existe e em U_1 tal que $S \cup \{e\} \in \mathcal{C}$ **faça**

3 $S \leftarrow S \cup \{e\}$

4 **devolva** S

GULOSO encontra um conjunto de peso maximal de $\mathcal{C}$.

Se $\mathcal{C}$ é um matroide então
todo conjunto produzido pelo GULOSO tem o mesmo peso.

Definição de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.
(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Se $\mathcal{C}$ é um matroide então
GULOSO sempre encontra um conjunto de $\mathcal{C}$ de peso máximo.

Definição de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.
(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Se $\mathcal{C}$ é um matroide então
GULOSO sempre encontra um conjunto de $\mathcal{C}$ de peso máximo.

$\mathcal{C}$ é um matroide se, para todo par A, B de conjuntos de $\mathcal{C}$ para os quais $|A| < |B|$, existe $e \in B \setminus A$ tal que $A \cup \{e\} \in \mathcal{C}$.

Definição de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.
(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Se $\mathcal{C}$ é um matroide então
GULOSO sempre encontra um conjunto de $\mathcal{C}$ de peso máximo.

$\mathcal{C}$ é um matroide se, para todo par A, B de conjuntos de $\mathcal{C}$ para os quais $|A| < |B|$, existe $e \in B \setminus A$ tal que $A \cup \{e\} \in \mathcal{C}$.

Essa é a definição do CLRS.

Exemplos de matroides

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Exemplos de matroides

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Dado um grafo G, a coleção de
todos os conjuntos de arestas de florestas de G é um matroide.

Exemplos de matroides

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Dado um grafo G, a coleção de
todos os conjuntos de arestas de florestas de G é um matroide.

Grafo bipartido $G = (U \cup V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo
uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

Exemplos de matroides

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Dado um grafo G, a coleção de
todos os conjuntos de arestas de florestas de G é um matroide.

Grafo bipartido $G = (U \cup V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

M_V: a coleção análoga com V no lugar de U.

Claro que M_V também é um matroide.

Exemplos de matroides

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Dado um grafo G, a coleção de
todos os conjuntos de arestas de florestas de G é um matroide.

Grafo bipartido $G = (U \cup V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

M_V: a coleção análoga com V no lugar de U.

Claro que M_V também é um matroide.

E $M_U \cap M_V$? É ou não é um matroide?

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

```
GULOSO (U, w, C)
1   (e_1, ..., e_n) ← ORDENE(U, w)   ▷ ordem decrescente dos pesos
2   S ← ∅
3   para i ← 1 até n faça
4         se S ∪ {e_i} ∈ C
5               então S ← S ∪ {e_i}
6   devolva S
```

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

```
GULOSO (U, w, C)
1   (e_1, ..., e_n) ← ORDENE(U, w)   ▷ ordem decrescente dos pesos
2   S ← ∅
3   para i ← 1 até n faça
4         se S ∪ {e_i} ∈ C
5               então S ← S ∪ {e_i}
6   devolva S
```

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo acima resolve o Problema 2.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Seja $A = (S \cap O) \cup \{e_i\}$. Note que $A \subseteq S$, logo $A \in \mathcal{C}$.
Seja A' um conjunto maximal de $\mathcal{C}$ tal que $A \subseteq A' \subseteq O \cup \{e_i\}$.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Seja $A = (S \cap O) \cup \{e_i\}$. Note que $A \subseteq S$, logo $A \in \mathcal{C}$.
Seja A' um conjunto maximal de $\mathcal{C}$ tal que $A \subseteq A' \subseteq O \cup \{e_i\}$.

Como $\mathcal{C}$ é um matroide, $|A'| = |O|$.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Seja $A = (S \cap O) \cup \{e_i\}$. Note que $A \subseteq S$, logo $A \in \mathcal{C}$.
Seja A' um conjunto maximal de $\mathcal{C}$ tal que $A \subseteq A' \subseteq O \cup \{e_i\}$.

Como $\mathcal{C}$ é um matroide, $|A'| = |O|$.
Logo $A' = O \setminus \{e_j\} \cup \{e_i\}$, com $j > i$.

De fato, como e_i é o primeiro a entrar em S de $S \setminus O$,
se $j < i$, então o elemento e_j teria sido incluído em S.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Seja $A = (S \cap O) \cup \{e_i\}$. Note que $A \subseteq S$, logo $A \in \mathcal{C}$.
Seja A' um conjunto maximal de $\mathcal{C}$ tal que $A \subseteq A' \subseteq O \cup \{e_i\}$.

Como $\mathcal{C}$ é um matroide, $|A'| = |O|$.
Logo $A' = O \setminus \{e_j\} \cup \{e_i\}$, com $j > i$.
Portanto o peso de A' é maior ou igual ao peso de O,
ou seja, A' tem peso máximo.

Prova do teorema

Teorema: Se $\mathcal{C}$ é um matroide,
então o algoritmo anterior resolve o Problema 2.

Prova:
Seja O um conjunto de $\mathcal{C}$ de peso máximo.
Seja S o conjunto devolvido pelo algoritmo.

Se $O = S$, não há nada a provar. Senão, note que $S \not\subseteq O$.

Seja i o menor possível tal que $e_i \in S \setminus O$.

Seja $A = (S \cap O) \cup \{e_i\}$. Note que $A \subseteq S$, logo $A \in \mathcal{C}$.
Seja A' um conjunto maximal de $\mathcal{C}$ tal que $A \subseteq A' \subseteq O \cup \{e_i\}$.

Como $\mathcal{C}$ é um matroide, $|A'| = |O|$.
Logo $A' = O \setminus \{e_j\} \cup \{e_i\}$, com $j > i$.
Portanto o peso de A' é maior ou igual ao peso de O,
ou seja, A' tem peso máximo.

Repetindo esse processo, provamos que S e O têm o mesmo peso.

Consequência nos exemplos

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Se cada vetor tiver um peso, podemos encontrar,
com o algoritmo guloso, uma base de peso máximo.

Consequência nos exemplos

Dado um subconjunto finito de um espaço vetorial, a coleção de todos os conjuntos de vetores LI deste subconjunto é um matroide.

Se cada vetor tiver um peso, podemos encontrar,
com o algoritmo guloso, uma base de peso máximo.

Dado um grafo G, a coleção de
todos os conjuntos de arestas de florestas de G é um matroide.

Se cada aresta tiver um peso, podemos encontrar,
com o algoritmo guloso, uma floresta de peso máximo
(Borůvka/Kruskal).

Clustering

U: conjunto de n itens $\{p_1, \ldots, p_n\}$.

$d(p_i, p_j)$: distância entre p_i e p_j.

- $d(p_i, p_i) = 0$ para $i = 1, \ldots, n$
- $d(p_i, p_j) = d(p_j, p_i) > 0$ para $i \neq j$

Clustering

U: conjunto de n itens $\{p_1, \ldots, p_n\}$.

$d(p_i, p_j)$: distância entre p_i e p_j.

- $d(p_i, p_i) = 0$ para $i = 1, \ldots, n$
- $d(p_i, p_j) = d(p_j, p_i) > 0$ para $i \neq j$

k: número de grupos.

k-clustering: partição de U em k partes.

Clustering

U: conjunto de n itens $\{p_1, \ldots, p_n\}$.

$d(p_i, p_j)$: distância entre p_i e p_j.

- $d(p_i, p_i) = 0$ para $i = 1, \ldots, n$
- $d(p_i, p_j) = d(p_j, p_i) > 0$ para $i \neq j$

k: número de grupos.

k-clustering: partição de U em k partes.

espaçamento de k-clustering: distância mínima
entre dois elementos de partes distintas do clustering.

Clustering

U: conjunto de n itens $\{p_1, \ldots, p_n\}$.

$d(p_i, p_j)$: distância entre p_i e p_j.

- $d(p_i, p_i) = 0$ para $i = 1, \ldots, n$
- $d(p_i, p_j) = d(p_j, p_i) > 0$ para $i \neq j$

k: número de grupos.

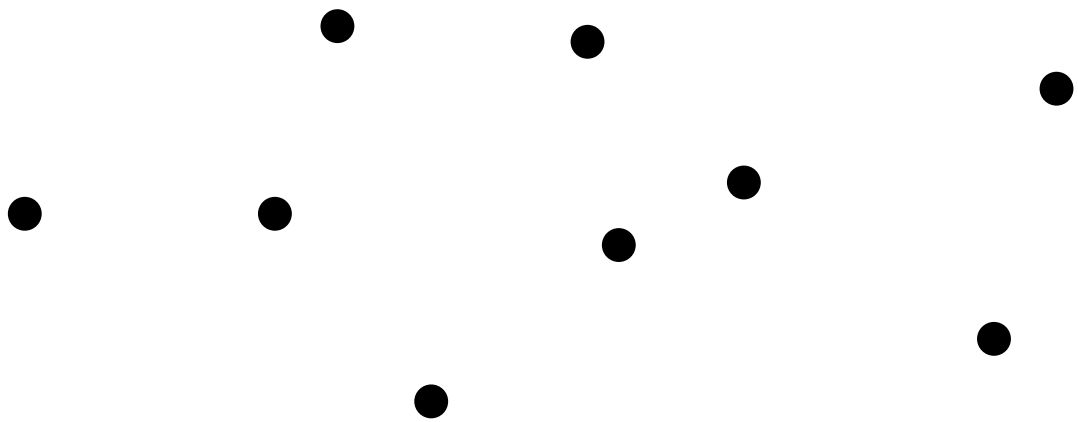
k-clustering: partição de U em k partes.

espaçamento de k-clustering: distância mínima
entre dois elementos de partes distintas do clustering.

Problema: Dados U, d e k,
encontrar um k-clustering com espaçamento máximo.

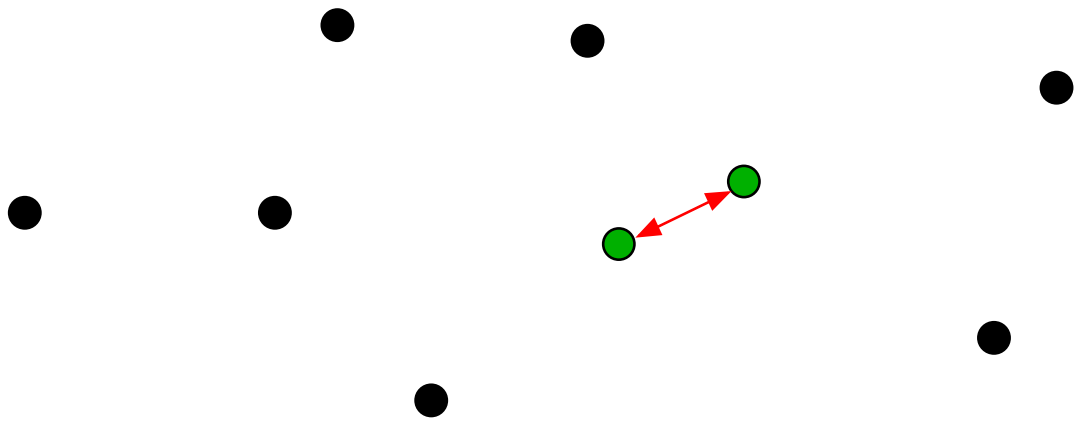
Clustering

Problema: Dados U, d e k,
encontrar um k-clustering com espaçamento máximo.



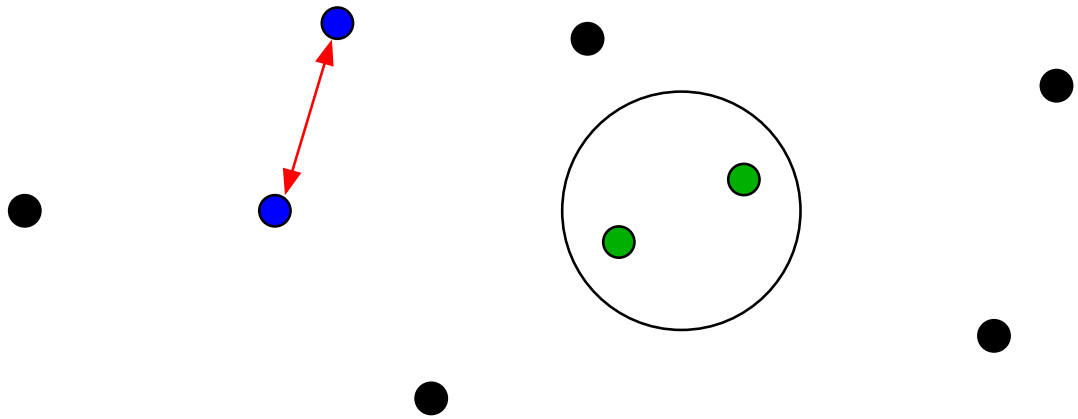
Clustering

Problema: Dados U, d e k,
encontrar um k-clustering com espaçamento máximo.



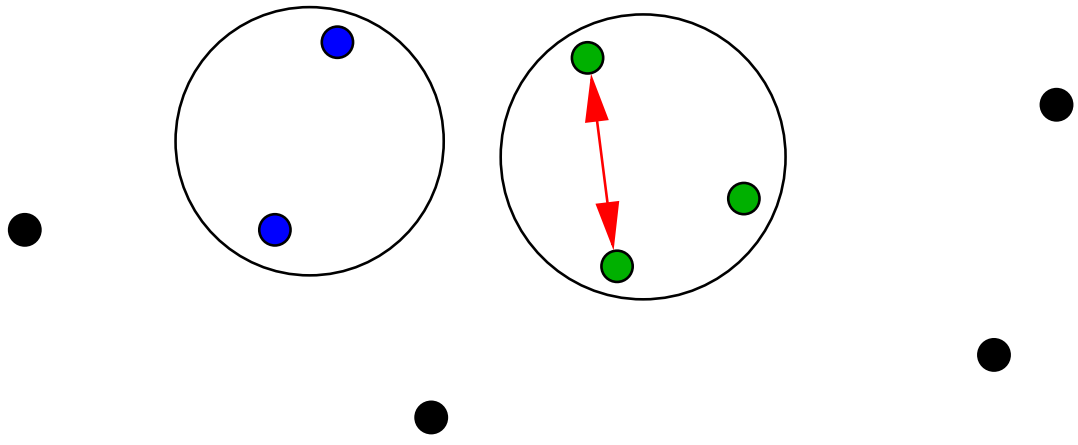
Clustering

Problema: Dados U, d e k,
encontrar um k-clustering com espaçamento máximo.



Clustering

Problema: Dados U, d e k,
encontrar um k-clustering com espaçamento máximo.



Algoritmo inspirado no Kruskal

Como adaptar MST-KRUSKAL para resolver o k-clustering?

Algoritmo inspirado no Kruskal

Como adaptar MST-KRUSKAL para resolver o k-clustering?

Acrescentamos arestas até termos k componentes.

Algoritmo inspirado no Kruskal

Como adaptar MST-KRUSKAL para resolver o k-clustering?

Acrescentamos arestas até termos k componentes.

CLUSTERING-KRUSKAL (V, E, c, k)

```
1  e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2  F ← ∅        i ← 0
3  enquanto G' = (V, F) tem > k componentes faça
4        i ← i + 1
5        se F ∪ {e_i} não tem circuito
6              então F ← F ∪ {e_i}
7  devolva F
```

Algoritmo inspirado no Kruskal

Como adaptar MST-KRUSKAL para resolver o k-clustering?

Acrescentamos arestas até termos k componentes.

CLUSTERING-KRUSKAL (V, E, c, k)

```
1  e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2  F ← ∅        i ← 0
3  enquanto G' = (V, F) tem > k componentes faça
4        i ← i + 1
5        se F ∪ {e_i} não tem circuito
6              então F ← F ∪ {e_i}
7  devolva F
```

Este algoritmo resolve o problema?

Análise da correção

CLUSTERING-KRUSKAL (V, E, c, k)

```
1   e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2   F ← ∅        i ← 0
3   enquanto G' = (V, F) tem ≥ k componentes faça
4         i ← i + 1
5         se F ∪ {e_i} não tem circuito
6               então F ← F ∪ {e_i}
7   devolva as componentes de F
```

Qual é o espaçamento do clustering devolvido?

Análise da correção

CLUSTERING-KRUSKAL (V, E, c, k)

```
1   e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2   F ← ∅        i ← 0
3   enquanto G' = (V, F) tem ≥ k componentes faça
4         i ← i + 1
5         se F ∪ {e_i} não tem circuito
6               então F ← F ∪ {e_i}
7   devolva as componentes de F
```

Qual é o espaçamento do clustering devolvido?

É o custo da próxima aresta que seria incluída em F.

Análise da correção

CLUSTERING-KRUSKAL (V, E, c, k)

```
1  e_1, ..., e_m ← ORDENE(E, c)          ▷ pelo valor de c_e
2  F ← ∅        i ← 0
3  enquanto G' = (V, F) tem ≥ k componentes faça
4        i ← i + 1
5        se F ∪ {e_i} não tem circuito
6              então F ← F ∪ {e_i}
7  devolva as componentes de F
```

Qual é o espaçamento do clustering devolvido?

É o custo da próxima aresta que seria incluída em F.

A próxima aresta que não forma circuito com F.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

O espaçamento de $\mathcal{C}$ é c_e.

Basta mostrar que o espaçamento de $\mathcal{C}^*$ é no máximo c_e.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

O espaçamento de $\mathcal{C}$ é c_e.

Basta mostrar que o espaçamento de $\mathcal{C}^*$ é no máximo c_e.

Ou $\mathcal{C} = \mathcal{C}^*$, ou algum C_i não está contido em nenhum C_j^*.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

O espaçamento de $\mathcal{C}$ é c_e.

Basta mostrar que o espaçamento de $\mathcal{C}^*$ é no máximo c_e.

Ou $\mathcal{C} = \mathcal{C}^*$, ou algum C_i não está contido em nenhum C_j^*.

Seja j tal que $C_i \cap C_j^* \neq \emptyset$.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

O espaçamento de $\mathcal{C}$ é c_e.

Basta mostrar que o espaçamento de $\mathcal{C}^*$ é no máximo c_e.

Ou $\mathcal{C} = \mathcal{C}^*$, ou algum C_i não está contido em nenhum C_j^*.

Seja j tal que $C_i \cap C_j^* \neq \emptyset$.

Seja f aresta de F em C_i com um único extremo em C_j^*.

Análise da correção

Seja e a próxima aresta que seria incluída em F.

Seja $\mathcal{C} = \{C_1, \ldots, C_k\}$ o k-clustering devolvido,
e $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$ um k-clustering ótimo.

O espaçamento de $\mathcal{C}$ é c_e.

Basta mostrar que o espaçamento de $\mathcal{C}^*$ é no máximo c_e.

Ou $\mathcal{C} = \mathcal{C}^*$, ou algum C_i não está contido em nenhum C_j^*.

Seja j tal que $C_i \cap C_j^* \neq \emptyset$.

Seja f aresta de F em C_i com um único extremo em C_j^*.

Claro que $c_f \leq c_e$ e espaçamento de $\mathcal{C}^*$ é no máximo c_f. □