

1. Árbol Generador de Peso Mínimo

1.1. Forma canónica o estándar

1.1.1. Ejemplar genérico

Una gráfica no dirigida $G = (V, E)$ donde V es el conjunto de vértices de la gráfica y E es el conjunto de aristas de la gráfica y cada arista tiene asociado un peso el cual es un número w tal que $w \in \mathbb{R}^+$ y un entero positivo k .

1.1.2. Pregunta de decisión

¿ G tiene un árbol generador tal que la suma de los pesos de sus aristas es menor o igual a k ?

1.2. Algoritmo no-determinístico polinomial

1.2.1. Fase adivinadora

Crearemos una gráfica $T = (V_T, E_T)$ tal que:

- $V_T = V$, es decir, la gráfica T tendrá a los mismos vértices que la gráfica original.
- $|E_T| = |V| - 1$, y además las aristas de E_T se formarán de manera no determinista.

1.2.2. Fase verificadora

- Verificaremos que la gráfica T sea un conexa sí es el caso entonces T es un árbol (*porque ya sabemos que $|E_T| = |V_T| - 1$*).
 - En éste punto tenemos que checar que $E_T \subseteq E$, ya que como creamos todas las aristas de E_T de manera no determinista, se puede dar el caso de que alguna arista $e \in E_T$ no pertenezca a la gráfica original, entonces en caso de que se cumple que para toda $e \in E_T$ también sucede que $e \in E$, verificamos que el peso del árbol T sea menor o igual que k , en caso verdadero **el candidato a solución** es una **solución al ejemplar** en otro caso **el candidato a solución** no es una **solución al ejemplar**.
 - Sí $E_T \not\subseteq E$ **el candidato a solución** no es una **solución al ejemplar**.
- En caso de que no sea conexa entonces T no es un árbol y por lo tanto no es una solución para el ejemplar.

1.3. Capturas de pantalla

```
src: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/HernandezJassoFaustoDavidR01/src$ python3 Reposicion.py
Ejemplar:
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16}
Aristas:
(v1, v5) Peso: 3, (v10, v15) Peso: 5, (v9, v1) Peso: 6, (v5, v10) Peso: 7, (v8, v1) Peso: 4, (v8, v0) Peso: 8, (v9, v11) Peso: 6, (v0, v7) Peso: 3, (v16, v9) Peso: 7, (v1, v2) Peso: 8, (v15, v9) Peso: 5, (v6, v16) Peso: 3, (v8, v10) Peso: 4, (v14, v16) Peso: 10, (v15, v16) Peso: 4, (v7, v8) Peso: 10, (v12, v14) Peso: 10, (v4, v8) Peso: 10, (v16, v3) Peso: 5, (v13, v6) Peso: 5, (v2, v14) Peso: 2, (v15, v0) Peso: 5, (v2, v3) Peso: 2, (v5, v13) Peso: 1, (v6, v7) Peso: 6, (v6, v8) Peso: 1, (v7, v2) Peso: 2, (v8, v11) Peso: 7, (v15, v3) Peso: 1, (v9, v5) Peso: 6, (v0, v2) Peso: 1, (v16, v11) Peso: 10, (v8, v5) Peso: 2, (v4, v15) Peso: 9, (v7, v5) Peso: 9, (v13, v10) Peso: 3, (v9, v6) Peso: 5, (v2, v10) Peso: 4, (v15, v2) Peso: 1
k = 62
Candidato a solución:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16}
Aristas:
{(v13, v16), (v14, v4), (v2, v11), (v2, v14), (v9, v7), (v15, v14), (v7, v10), (v8, v14), (v7, v6), (v16, v15), (v3, v16), (v9, v0), (v8, v10), (v5, v16), (v10, v9), (v15, v7)}
No es una solución para el ejemplar.
```

Figura 1: Captura de pantalla 1

```
src: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/HernandezJassoFaustoDavidR01/src$ python3 Reposicion.py
Ejemplar:
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14}
Aristas:
(v6, v0) Peso: 9, (v14, v5) Peso: 4, (v12, v1) Peso: 4, (v10, v12) Peso: 8, (v11, v14) Peso: 6, (v3, v9) Peso: 3, (v1, v2) Peso: 6, (v1, v7) Peso: 10, (v8, v5) Peso: 4, (v9, v0) Peso: 5, (v1, v0) Peso: 2, (v13, v1) Peso: 10, (v11, v9) Peso: 2, (v4, v2) Peso: 2, (v3, v1) Peso: 7, (v11, v1) Peso: 6, (v1, v6) Peso: 1, (v13, v7) Peso: 10, (v13, v11) Peso: 6, (v0, v3) Peso: 1, (v8, v13) Peso: 4, (v12, v0) Peso: 1, (v7, v6) Peso: 3, (v13, v3) Peso: 8, (v11, v7) Peso: 1, (v4, v0) Peso: 8, (v3, v7) Peso: 6, (v0, v14) ) Peso: 8, (v14, v4) Peso: 2, (v6, v3) Peso: 7, (v4, v6) Peso: 3, (v12, v8) Peso: 2, (v7, v10) Peso: 4, (v8, v1) Peso: 2, (v12, v9) Peso: 3, (v11, v4) Peso: 7, (v12, v11) Peso: 6, (v3, v14) Peso: 8, (v6, v5) Peso: 5, (v5, v11) Peso: 1, (v2, v9) Peso: 9, (v7, v8) Peso: 6, (v10, v6) Peso: 8, (v8, v6) Peso: 3, (v6, v2) Peso: 9, (v10, v11) Peso: 6, (v10, v8) Peso: 7, (v6, v9) Peso: 7, (v8, v9) Peso: 2, (v2, v13) Peso: 4, (v12, v5) Peso: 6, (v10, v3) Peso: 4, (v5, v9) Peso: 2, (v0, v2) Peso: 5, (v8, v2) Peso: 4, (v8, v11) Peso: 10
k = 61
Candidato a solución:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14}
Aristas:
{(v2, v14), (v11, v6), (v6, v5), (v6, v10), (v14, v1), (v8, v13), (v2, v13), (v6, v13), (v3, v1), (v11, v1), (v5, v0), (v1, v4), (v0, v12), (v0, v8)}
No es una solución para el ejemplar.
```

Figura 2: Captura de pantalla 2

```
src: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/HernandezJassoFaustoDavidR01/src$ python3 Reposicion.py
*****
Ejemplar 1
Gráfica
Vértices:
{v1, v2, v3, v4, v5, v6, v7, v8, v9}
Aristas:
{(v1, v2), (v1, v3), (v2, v4), (v2, v5), (v3, v5), (v3, v6), (v4, v5), (v4, v7), (v5, v7), (v6, v7), (v7, v8), (v8, v9), (v9, v6)}
Árbol Generador de Peso menor o igual a 19
Gráfica
Vértices:
{v1, v2, v3, v4, v5, v6, v7, v8, v9}
Aristas:
{(v1, v2), (v1, v3), (v2, v5), (v4, v5), (v5, v7), (v6, v7), (v7, v8), (v8, v9)}
*****
```

Figura 3: Captura de pantalla 3

```
src: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/HernandezJassoFaustoDavidR01/src$ python3 Reposicion.py
Ejemplar 2
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8}
Aristas:
{(v0, v1), (v0, v2), (v1, v2), (v1, v5), (v2, v3), (v2, v4), (v2, v5), (v3, v4), (v4, v5), (v5, v6), (v5, v7), (v6, v7), (v7, v8)}
Árbol Generador con peso menor o igual a 27
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8}
Aristas:
{(v0, v1), (v1, v2), (v1, v5), (v2, v3), (v2, v4), (v5, v6), (v6, v7), (v7, v8)}
*****
```

Figura 4: Captura de pantalla 4

```
src: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/HernandezJassoFaustoDavidR01/src$ python3 Reposicion.py
+++++
Ejemplar 3
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6}
Aristas:
{(v0, v1), (v0, v3), (v0, v4), (v1, v2), (v1, v4), (v1, v6), (v2, v6), (v3, v4), (v3, v5), (v4, v5), (v4, v6), (v5, v6)}
Árbol Generador con peso menor o igual a 14
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6}
Aristas:
{(v0, v3), (v0, v4), (v1, v2), (v1, v6), (v4, v5), (v4, v6)}
+++++
```

Figura 5: Captura de pantalla 5

2. Coloración de Gráficas

2.1. Forma canónica o estándar

2.1.1. Ejemplar genérico

Una gráfica no dirigida $G = (V, E)$ donde V es el conjunto de vértices de la gráfica y E es el conjunto de aristas de la gráfica y un entero positivo k tal que $k \leq |V|$.

2.1.2. Pregunta de decisión

¿ G tiene una k -coloración en los vértices?

2.2. Algoritmo no-determinístico polinomial

2.2.1. Fase adivinadora

- Para cada vértice v tal que $v \in V$ a v le asigna uno de los k colores distintos de manera no determinista.

2.2.2. Fase verificadora

- Para cada vértice v tal que $v \in V$ hacemos lo siguiente:
 - Para cada vecino u de v verificamos que el color de u y el color de v son distintos, aquí tenemos dos casos:
 - Sí el color de u es igual al de v entonces nuestro candidato a solución no es una solución.
 - Sí el color de u es distinto al de v entonces continuamos la verificación con el resto de los vecinos de v .
- En éste punto sabemos que para todo par de vertices que son vecinos en G , éstos tienen un color distinto entre sí, por lo tanto nuestro candidato a solución es una solución para el ejemplar.

2.3. Capturas de pantalla

```
lechedevainilla@lechedevainilla-67-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/src$ python3 Reposicion.py
Ejemplar:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9}
Aristas:
{(v7, v6), (v8, v1), (v4, v1), (v9, v4), (v3, v5), (v8, v0), (v5, v0), (v9, v2), (v2, v0), (v8, v6), (v4, v0), (v0, v1), (v4, v5), (v7, v4)}
k = 10
Candidato a solución:
Vértice: v0
Color: 4
Vértice: v1
Color: 9
Vértice: v2
Color: 6
Vértice: v3
Color: 3
Vértice: v4
Color: 8
Vértice: v5
Color: 5
Vértice: v6
Color: 5
Vértice: v7
Color: 1
Vértice: v8
Color: 6
Vértice: v9
Color: 3
El candidato sí es una solución
```

Figura 6: Captura de pantalla 1

```

lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/src$ python3 Reposicion.py
Ejemplar:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9}
Aristas:
{(v8, v4), (v9, v8), (v9, v1), (v0, v6), (v8, v1), (v3, v4), (v5, v0), (v6, v2), (v0, v7), (v2, v3), (v2, v1), (v8, v5), (v1, v7), (v6, v1), (v0, v3), (v3, v9), (v8, v7), (v7, v2)}
k = 10
Candidato a solución:
Vértice: v0
Color: 1
Vértice: v1
Color: 6
Vértice: v2
Color: 4
Vértice: v3
Color: 10
Vértice: v4
Color: 10
Vértice: v5
Color: 3
Vértice: v6
Color: 2
Vértice: v7
Color: 2
Vértice: v8
Color: 4
Vértice: v9
Color: 6
El candidato no es una solución

```

Figura 7: Captura de pantalla 2

```

lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/src$ python3 Reposicion.py
Ejemplar:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9}
Aristas:
{(v1, v9), (v7, v6), (v9, v4), (v9, v0), (v6, v2), (v4, v7), (v8, v4), (v2, v9), (v6, v3), (v2, v3), (v5, v8), (v4, v5)}
k = 10
Candidato a solución:
Vértice: v0
Color: 10
Vértice: v1
Color: 9
Vértice: v2
Color: 5
Vértice: v3
Color: 8
Vértice: v4
Color: 6
Vértice: v5
Color: 2
Vértice: v6
Color: 6
Vértice: v7
Color: 9
Vértice: v8
Color: 3
Vértice: v9
Color: 8
El candidato sí es una solución

```

Figura 8: Captura de pantalla 3

```

lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/src$ python3 Reposicion.py
Ejemplar:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14}
Aristas:
{(v9, v12), (v0, v7), (v12, v0), (v13, v6), (v7, v4), (v9, v0), (v4, v2), (v7, v11), (v7, v14), (v12, v5), (v10, v12), (v3, v13), (v11, v6), (v13, v10), (v9, v10), (v2, v1), (v11, v3), (v0, v8), (v9, v6), (v3, v8), (v8, v1), (v5, v1), (v11, v13), (v13, v0)}
k = 10
Candidato a solución:
Vértice: v0
Color: 10
Vértice: v1
Color: 5
Vértice: v2
Color: 9
Vértice: v3
Color: 5
Vértice: v4
Color: 2
Vértice: v5
Color: 3
Vértice: v6
Color: 1
Vértice: v7
Color: 5
Vértice: v8
Color: 7
Vértice: v9
Color: 9
Vértice: v10
Color: 1
Vértice: v11
Color: 8
Vértice: v12
Color: 8
Vértice: v13
Color: 3
Vértice: v14
Color: 10
El candidato sí es una solución

```

Figura 9: Captura de pantalla 4

```

lechedevainilla@lechedevainilla-G7-7588:~/Semestre2023-1/Complejidad_Computacional/Programas/Reposicion/src$ python3 Reposicion.py
Ejemplar:
Gráfica
Vértices:
{v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14}
Aristas:
{(v3, v6), (v6, v9), (v0, v13), (v14, v5), (v3, v1), (v11, v4), (v8, v6), (v6, v12), (v12, v3), (v5, v0), (v12, v9), (v12, v7), (v3, v7), (v10, v9), (v9, v7), (v12, v8), (v14, v8), (v12, v10), (v5, v10), (v1, v5), (v14, v13), (v5, v8), (v4, v14), (v14, v0), (v8, v0), (v10, v14), (v5, v12)}
k = 10
Candidato a solución:
Vértice: v0
Color: 3
Vértice: v1
Color: 6
Vértice: v2
Color: 1
Vértice: v3
Color: 2
Vértice: v4
Color: 9
Vértice: v5
Color: 9
Vértice: v6
Color: 3
Vértice: v7
Color: 9
Vértice: v8
Color: 7
Vértice: v9
Color: 8
Vértice: v10
Color: 3
Vértice: v11
Color: 5
Vértice: v12
Color: 10
Vértice: v13
Color: 5
Vértice: v14
Color: 6
El candidato sí es una solución

```

Figura 10: Captura de pantalla 5