EIF203 Quiz Kruskal

Análisis del algoritmo de Kruskal

Diego Quirós Artiñano
ID:901150326

EIF-203: Estructuras Discretas NRC: 41712
Universidad Nacional de Costa Rica

13 de junio, 2022

Declaración Jurada

Declaro de manera jurada que este trabajo fue elaborado por mi persona de manera estrictamente individual y que las fuentes usadas son la declaradas en la lámina de referencias, las cuales sirvieron de base pero no fueron usadas como copia literal.

Referencias I

Albertson, M. O. & Hutchinson, J. P. (1988). *Discrete mathematics with algorithms*. Nashville, TN, John Wiley & Sons.

algorithm - ¿Cuál es la diferencia entre un heurístico y un algoritmo? (2022). https://www.web-dev-qa-db-es.com/es/algorithm/cual-es-la-diferencia-entre-un-heuristico-y-un-algoritmo/968038045

Crisler, N., Froelich, G. & Fisher, P. (1999). *Discrete mathematics through applications* (2.ª ed.). New York, NY, W.H. Freeman.

Dasgupta, S., Papadimitriou, C. H. & Vazirani, U. V. (2006). *Algorithms*. New York, NY, McGraw-Hill Professional.

Disjoint Set (Or Union-Find) | Set 1 (Detect Cycle in an Undirected Graph) - GeeksforGeeks. (2022). https://www.geeksforgeeks.org/union-find

Goodaire, E. G. & Parmenter, M. M. (2001). *Discrete mathematics with graph theory* (2.ª ed.). Upper Saddle River, NJ, Pearson.

Referencias II

Kruskal's Minimum Spanning Tree Algorithm | Greedy Algo-2 - GeeksforGeeks. (2022). https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2

Tabla de Contenidos

Definiciones y explicaciones

Minimum Spanning Tree o árbol de exploración mínimo

Spanning Tree

Según Goodaire y Parmenter, 2001, es un subgrafo de un grafo conectado que incluye todos los vertices

Minimum Spanning Tree

Es un subgrafo de un grafo con peso, pero con el mínimo peso posible

Heurístico y Greedy

Algoritmo Heurístico: Como visto en clase el algoritmo heurístico es el algoritmo que va haciendo decisiones sin conocimiento de lo siguiente. O como dicen en este foro («algorithm - ¿Cuál es la diferencia entre un heurístico y un algoritmo?», 2022), es "Las heurísticas implican utilizar un enfoque de aprendizaje y descubrimiento para alcanzar una solución"

Algoritmo Greedy: Como se vio en clase el algoritmo greedy es un algoritmo heurístico que no hace backtracking.

Algoritmo de Kruskal

El algoritmo de Kruskal es un algoritmo heurístico y greedy para buscar el árbol de exploración de peso mínimo

Análisis del algoritmo de Kruskal

Algoritmo de Kruskal I

Algoritmo 1 Algoritmo de Kruskal

$V \leftarrow$ `list(vertices)`
$E \leftarrow$ `list(aristasConPesos)`
$T = []$
if $0 \leq \texttt{len}(E) \leq \texttt{len}(V) \wedge \texttt{len}(V) > 0$ **then** ▷ Sacado de Albertson y Hutchinson, 1988 que tiene que se V-1 y el paso 1 de Crisler y col., 1999
    return Error: no hay aristas o no está conectado
end if

Algoritmo de Kruskal II

```
sort(E)        ▷ ajusta para que los vertices estén ordenados acendientemente por peso, no explicitamente mencionado en todos los libros, pero Albertson y Hutchinson, 1988 si lo toma en cuenta a la hora de hacer el cálculo de tiempo
for i in E do
    if !find(i, T) then        ▷ busca que no haya i en T y que i no haya un ciclo con lo que ya está guardando en T
        union(i, T)        ▷ ingresa la arista i
    end if
end for
```

Tiempo del algoritmo de Kruskal I

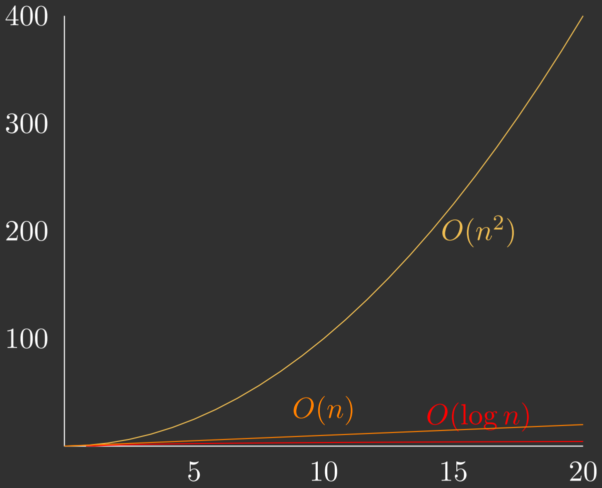
Aquí podemos asimilar varios análisis de diferentes fuentes.

- Primero Albertson y Hutchinson, 1988 concluye que si el sorteo de las aristas no está optimizado puede llegar a ser $O(E^2)$. Las comparaciones (lo que están tomando como el find-union, o disjoint-set como lo conocen en otros libros, en este solo mencionan una unión y comparación) puede llegar a ser hasta $O(V)$ comparaciones o a lo máximo $O(EV)$. Entonces que la complejidad de tiempo es de $O(E^2 + EV)$ o con un algoritmo de sorteo eficiente $O(E \log E)$ (E siendo el número de aristas y V siendo el número de vertices). Y por último menciona que $E \leq V(V-1)/2 = O(V^2)$

Tiempo del algoritmo de Kruskal II

- Segundo tanto «Kruskal's Minimum Spanning Tree Algorithm | Greedy Algo-2 - GeeksforGeeks», 2022 como Dasgupta y col., 2006 concluyen que sabiendo $\log E \approx \log V$ el resultado es $O(E \log E + E \log E)$ (uno el tiempo del sorteo el otro como el tiempo del union-find) o también se puede expresar como $O(E \log E)$ (o $O(E \log V)$)
- Tercero es importante notar que «Disjoint Set (Or Union-Find) | Set 1 (Detect Cycle in an Undirected Graph) - GeeksforGeeks», 2022 prueba que el algoritmo de union() y find() se puede hacer con fuerza bruta y da en el peor de los casos $O(n)$ (lineal), verificando lo que dice la primera fuente que es el número de aristas * números de vertices
- Entonces para concluir como el curso es pesimista en su análisis el peor resultado es $O(E^2) + O(EV) \sim O(E^2)$, en vez de $O(E \log E) + O(E \log E) \sim O(E \log E)$

Tiempo del algoritmo de Kruskal III



Prueba que $O(E^2)$ es el de mayor impacto

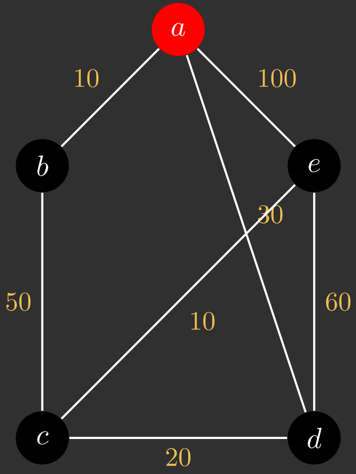
Tiempo del algoritmo de Kruskal IV

Resumen de puntos importantes

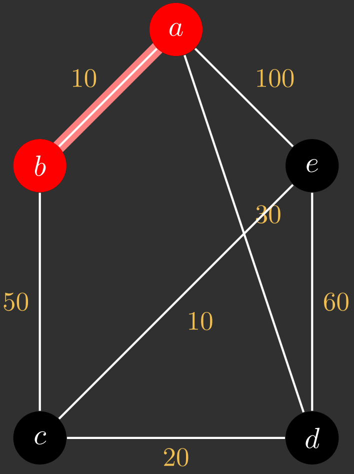
Tiempo de sorteo pesimista: $O(E^2)$
Tiempo del find-union pesimista: $O(EV)$
Tiempo de Kruskal Pesimista: $O(E^2) + O(EV) \sim O(E^2)$
Tiempo de Kruskal optimizado: $O(E \log E) + O(E \log E) \sim O(E \log E)$

Ejemplos

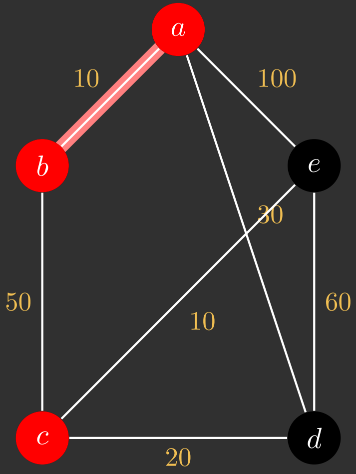
Ejemplo 1 (Clase 6 de junio)



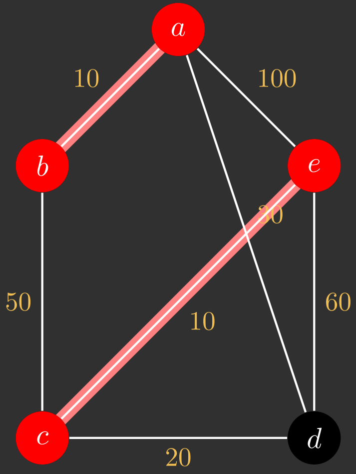
Ejemplo 1 (Clase 6 de junio)



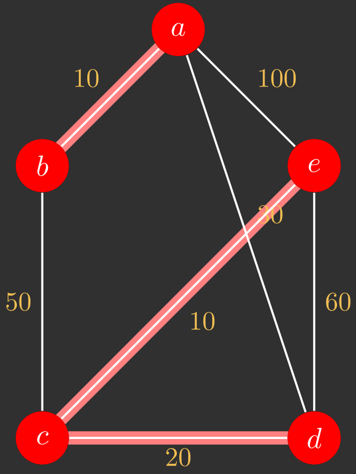
Ejemplo 1 (Clase 6 de junio)



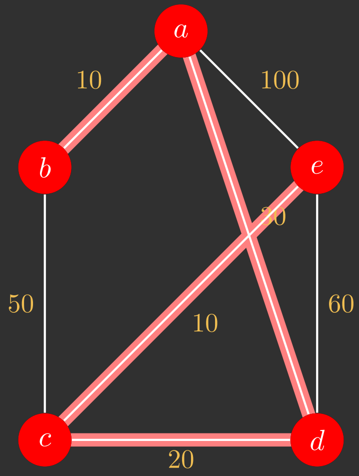
Ejemplo 1 (Clase 6 de junio)



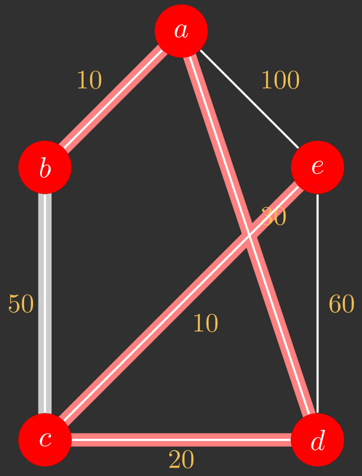
Ejemplo 1 (Clase 6 de junio)



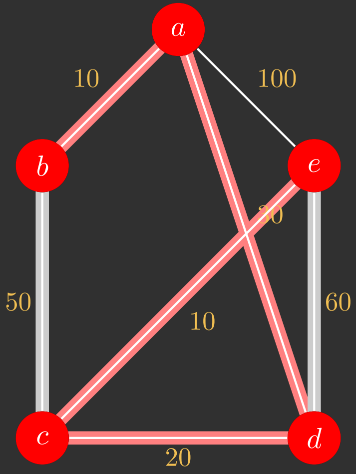
Ejemplo 1 (Clase 6 de junio)



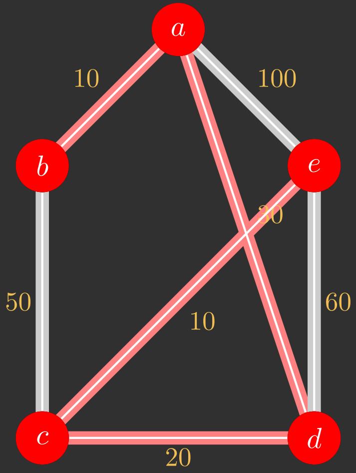
Ejemplo 1 (Clase 6 de junio)



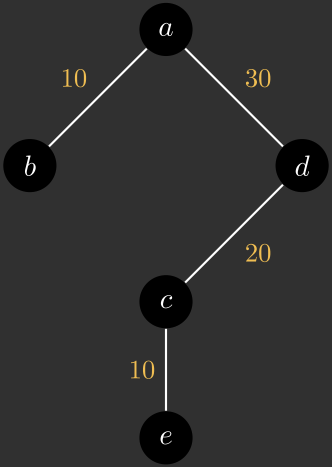
Ejemplo 1 (Clase 6 de junio)



Ejemplo 1 (Clase 6 de junio)

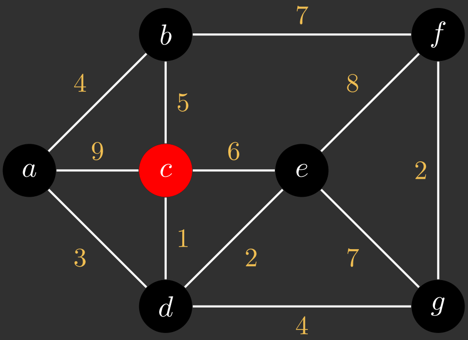


Resultado Ejemplo 1

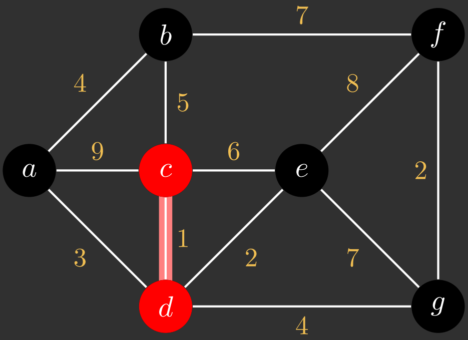


Peso del árbol: 70

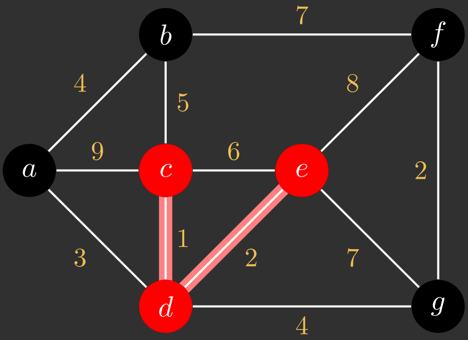
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



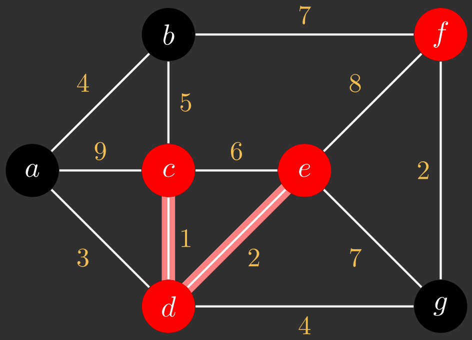
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



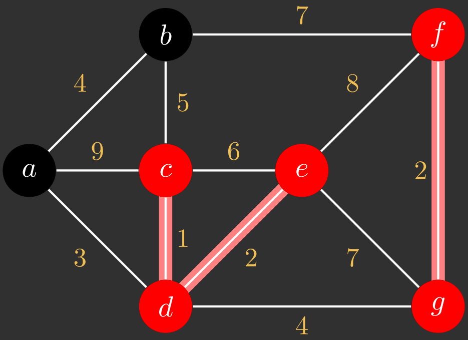
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



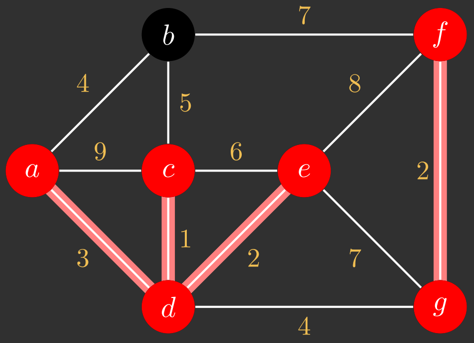
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



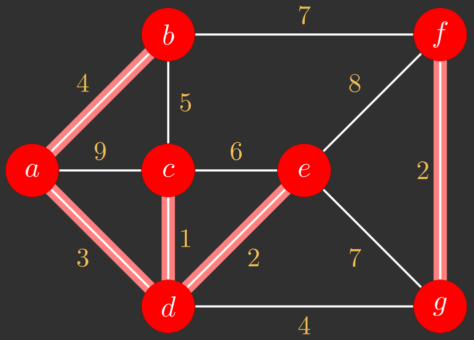
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



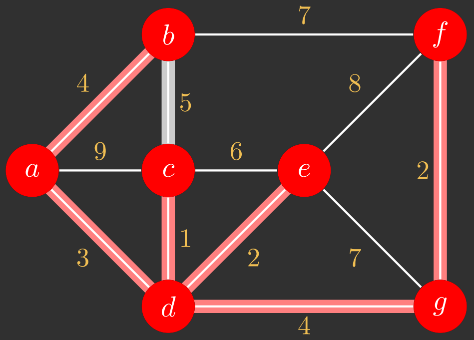
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



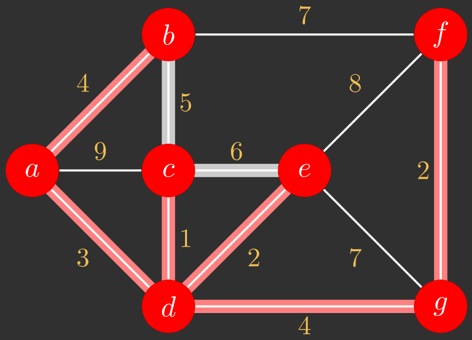
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



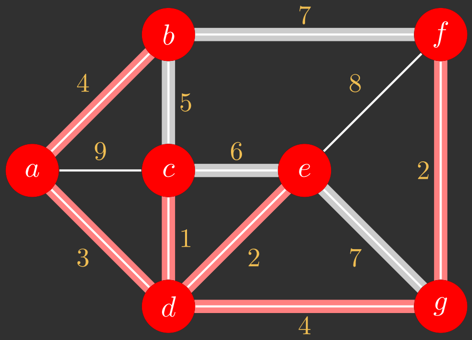
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



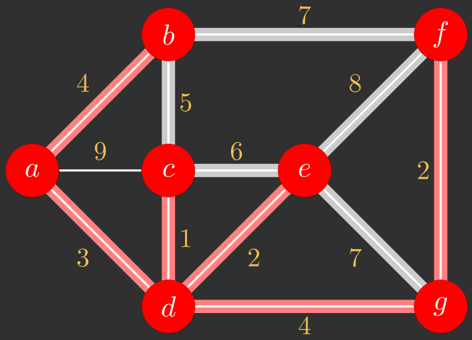
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



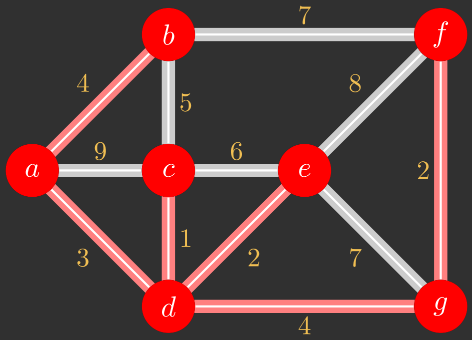
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



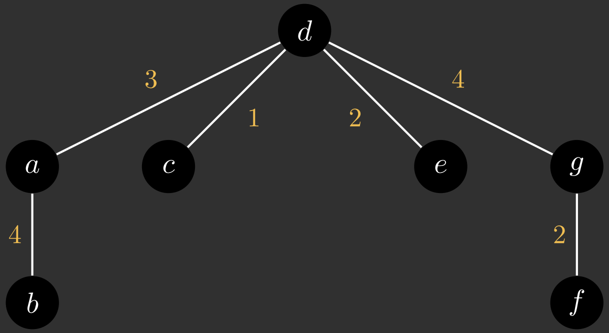
Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



Ejemplo 2 (Ejercicio 6, Práctica Parcial 2)



Resultado Ejemplo 2



Peso del árbol: 16