

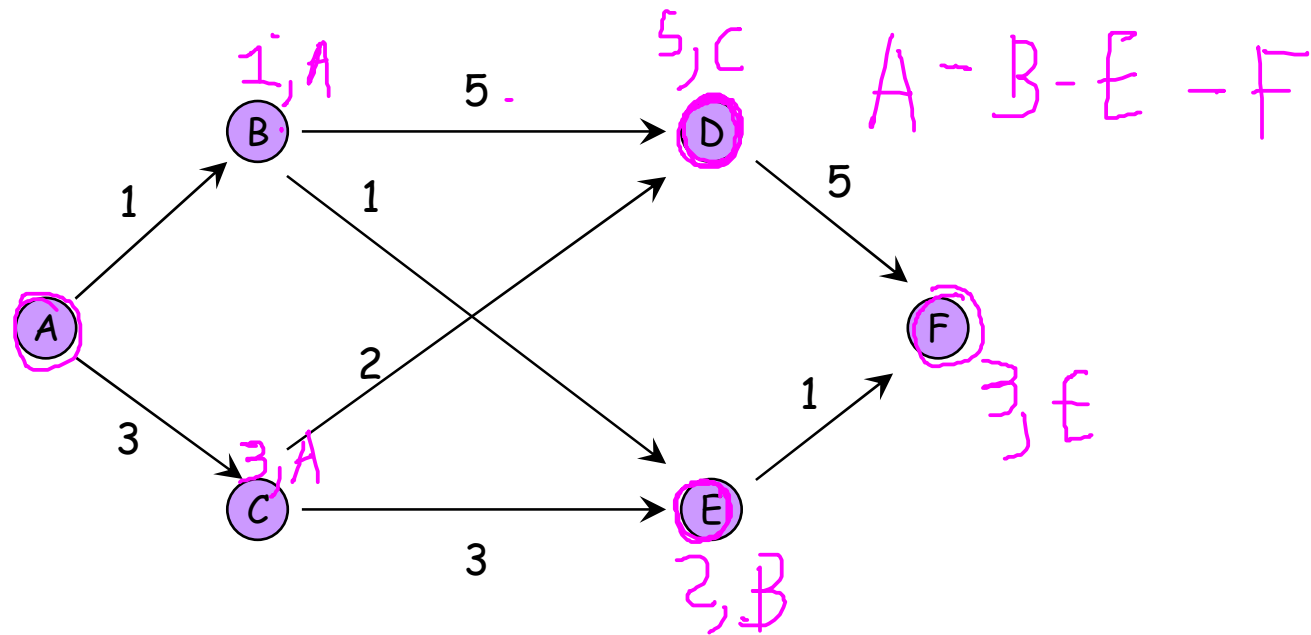
Fundamentos de análisis y diseño de algoritmos

Programación voraz

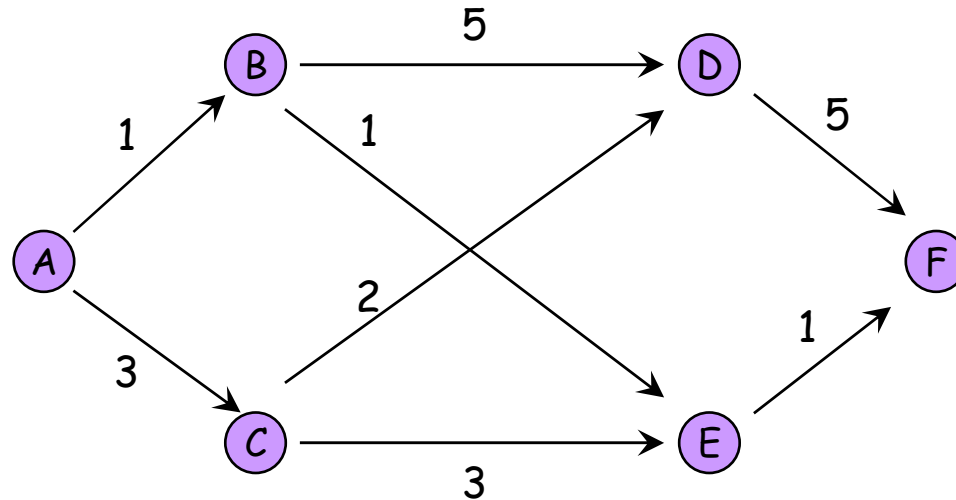
~~X~~ Camino más corto 1 a muchos

Camino más corto muchos a muchos

Programación voraz

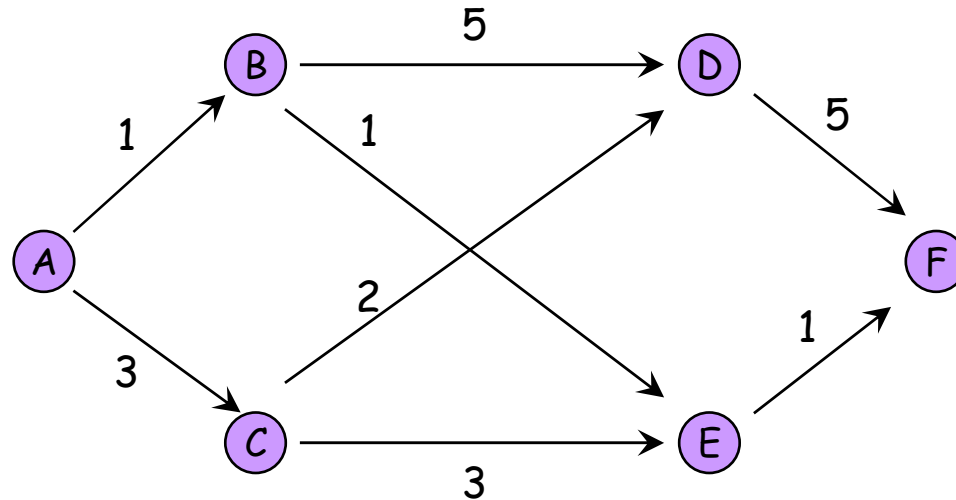


Programación voraz



Problema: encontrar el camino de menor costo entre un vértice s y todos los demás vértices

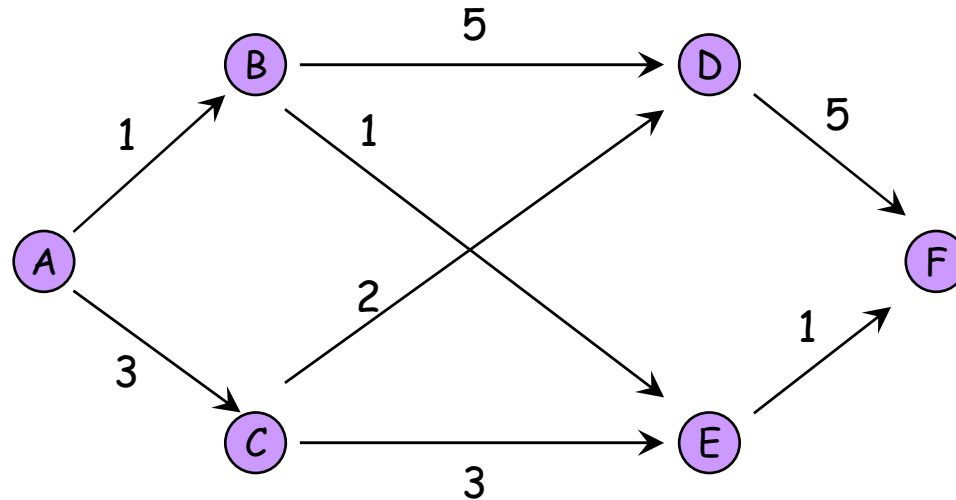
Programación voraz



Dijkstra(G, w, s)

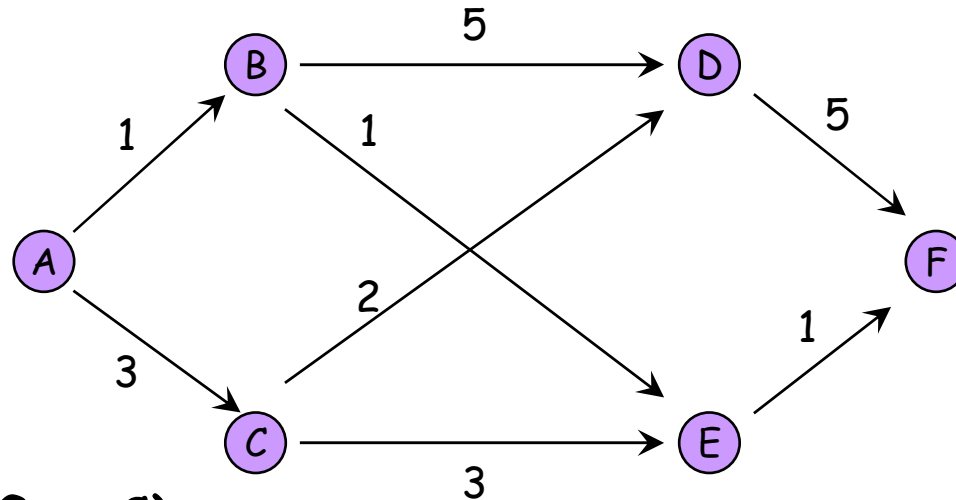
- $G = (V, E)$
- w es una función que asigna los pesos a las aristas
- s es el vértice

Programación voraz



Dijkstra(G, w, C)

Programación voraz

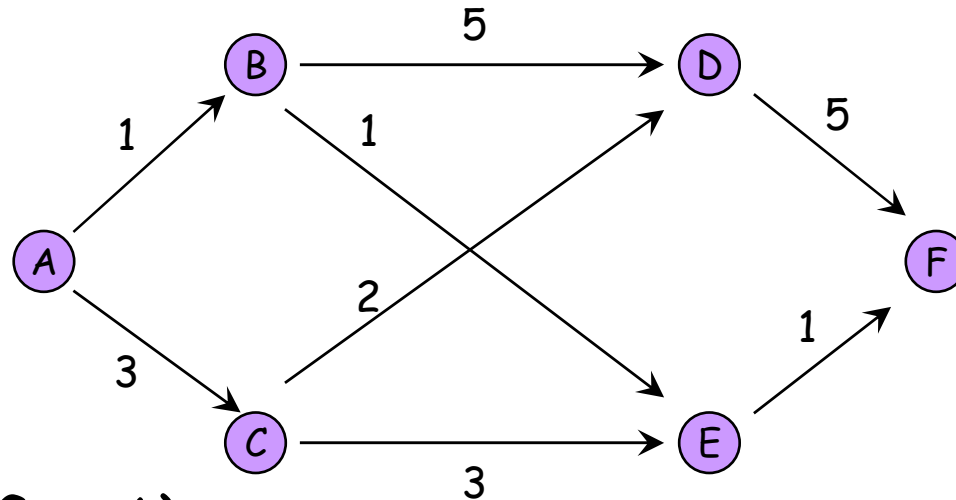


Dijkstra(G, w, C)

	A	B	C	D	E	F
d	∞	∞	0	2	3	4

	A	B	C	D	E	F
π	nil	nil	nil	C	C	E

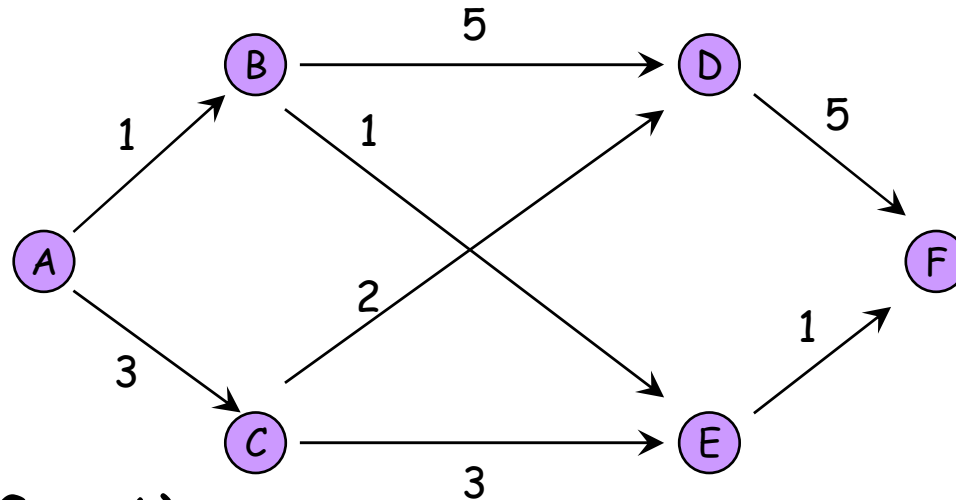
Programación voraz



Dijkstra(G, w, A)

	A	B	C	D	E	F
d						
π						

Programación voraz



Dijkstra(G, w, A)

	A	B	C	D	E	F
d	0	1	3	5	2	3

	A	B	C	D	E	F
π	nil	A	A	C	B	E

Programación voraz

- Dado un grafo dirigido $G=(V,E)$, el peso de un camino $p=\langle v_0, v_1, v_2, \dots, v_k \rangle$ es la suma de los pesos de las aristas que lo constituyen

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

Programación voraz

Problema del camino más corto de un origen a muchos destinos:

- Dado un grafo $G=(V,E)$, encontrar un camino más corto desde un vértice origen ($s \in V$) a cada vértice $v \in V$

Programación voraz

Estrategia voraz: de las aristas de un vértice v , seleccionar aquella con menor peso

Programación voraz

INITIALIZE-SINGLE-SOURCE(G, s)

foreach $v \in V[G]$

do $d[v] \leftarrow \infty$

$\pi[v] \leftarrow \text{nil}$

$d[s] \leftarrow 0$

Programación voraz

INITIALIZE-SINGLE-SOURCE(G, s)

foreach $v \in V[G]$

do $d[v] \leftarrow \infty$

$\pi[v] \leftarrow \text{nil}$

$d[s] \leftarrow 0$

$$T = \Theta(|V|)$$

DIJKSTRA(G, s)

INITIALIZE-SINGLE-SOURCE(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V[G]$

while $Q \neq \{ \}$

do $u \leftarrow \text{EXTRACT-MIN}(Q)$

$S \leftarrow S \cup \{u\}$

foreach $v \in \text{Adj}[u]$

do if $d[v] > d[u] + w(u, v)$

then $d[v] = d[u] + w(u, v)$

$\pi[v] \leftarrow u$

DIJKSTRA(G, s)

INITIALIZE-SINGLE-SOURCE(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V[G]$

while $Q \neq \{ \}$

do $u \leftarrow \text{EXTRACT-MIN}(Q)$

$S \leftarrow S \cup \{u\}$

foreach $v \in \text{Adj}[u]$

do if $d[v] > d[u] + w(u, v)$

then $d[v] = d[u] + w(u, v)$

$\pi[v] \leftarrow u$

Q es una cola de prioridad, donde la prioridad es el valor en d. Se extrae el vértice con menor valor

DIJKSTRA(G, s)

INITIALIZE-SINGLE-SOURCE(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V[G]$

$$T = \Theta(|V|^2)$$

while $Q \neq \{ \}$

do $u \leftarrow \text{EXTRACT-MIN}(Q)$

$S \leftarrow S \cup \{u\}$

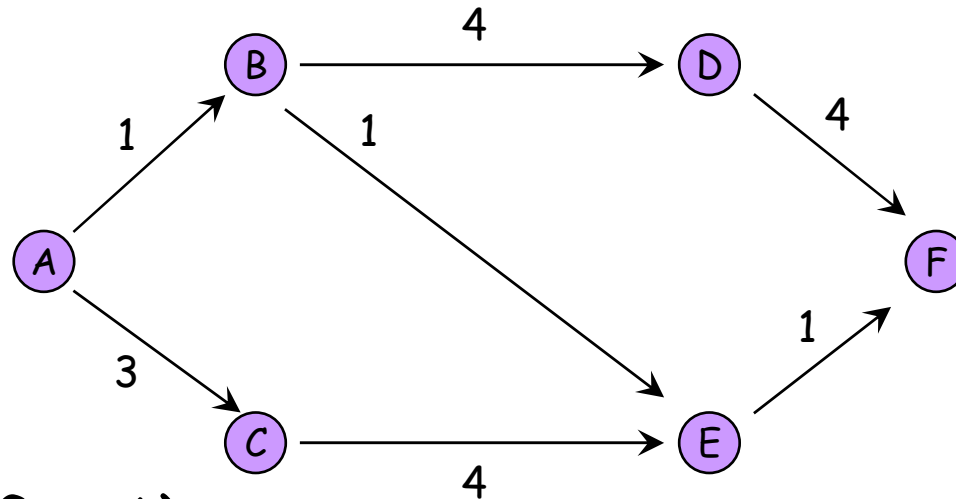
foreach $v \in \text{Adj}[u]$

do if $d[v] > d[u] + w(u, v)$

then $d[v] = d[u] + w(u, v)$

$\pi[v] \leftarrow u$

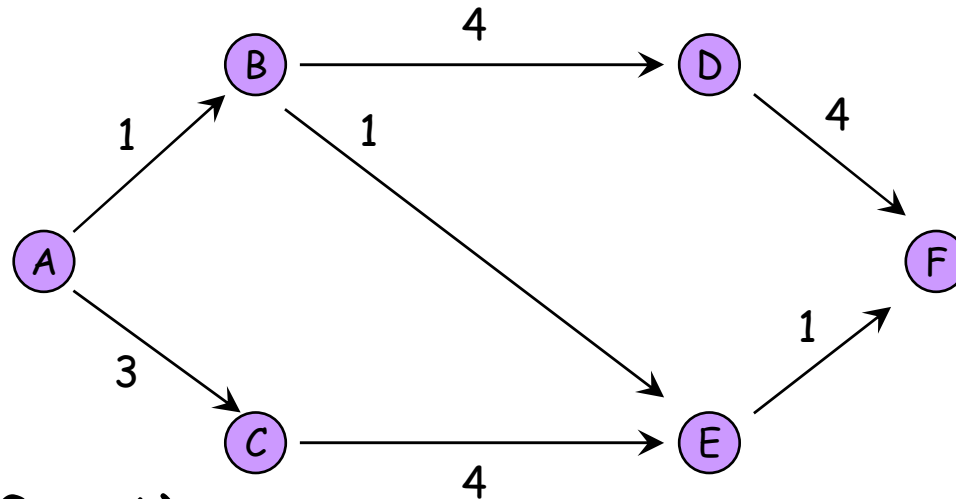
Programación voraz



Dijkstra(G, w, A)

	A	B	C	D	E	F
d						
π						

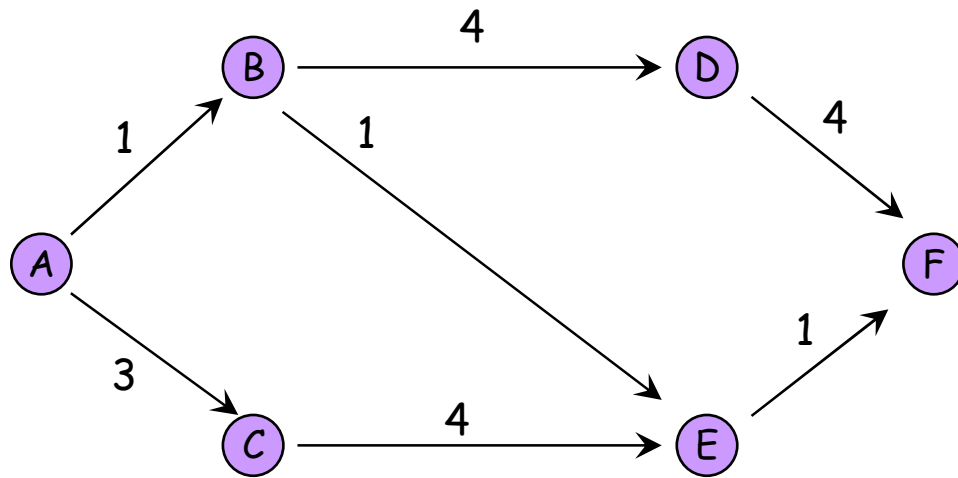
Programación voraz



Dijkstra(G, w, A)

	A	B	C	D	E	F
d	0	∞	∞	∞	∞	∞

	A	B	C	D	E	F
π	nil	nil	nil	nil	nil	nil



Dijkstra(G, w, A)

DIJKSTRA(G, s)

INITIALIZE-SINGLE-SOURCE(G, s)

$S \leftarrow \{\}$

$Q \leftarrow V[G]$

while $Q \neq \{\}$

do $u \leftarrow \text{EXTRACT-MIN}(Q)$

$S \leftarrow S \cup \{u\}$

foreach $v \in \text{Adj}[u]$

do if $d[v] > d[u] + w(u, v)$

then $d[v] = d[u] + w(u, v)$

$\pi[v] \leftarrow u$

A	B	C	D	E	F
0	∞	∞	∞	∞	∞

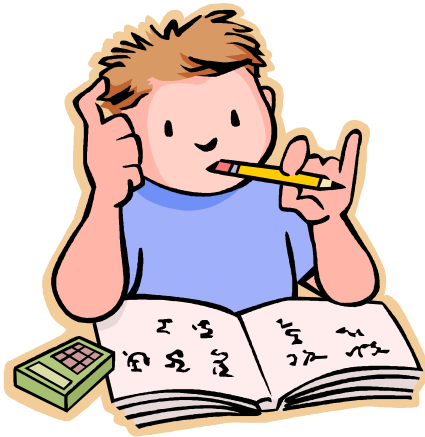
A	B	C	D	E	F
nil	nil	nil	nil	nil	nil



Listar todas las posibles soluciones
y escoger la mejor

Programación ingenua

Mirar como se divide el problema
Identificar la subestructura optima
Obtener solución optima (Garantizado)

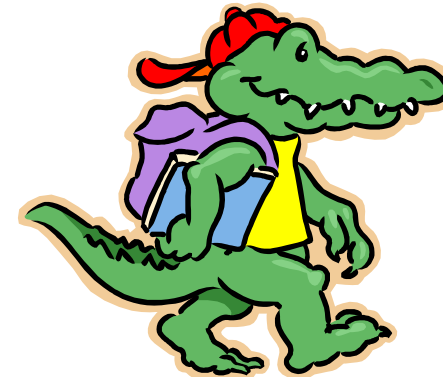


Programación dinámica



Mirar como se divide el problema
PERO, se toma una decisión
LOCAL

Rapido -- Barato pero no es
optimo (Bonito)



Programación voraz

EVALUACIÓN DE CURSOS / Universidad del Valle (Cali, Colombia) - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

https://swebse12.univalle.edu.co/evaluaciondocente/ Ir

Primeros pasos Últimas noticias

Universidad del Valle

Cali - Colombia Inicio | Buscar | Directorio | Biblioteca


Facultades y Dependencias / Vicerrectoría Académica

EVALUACIÓN DE CURSOS

Importancia de la evaluación

Evalúe los cursos y el desempeño de los profesores.
Opine sobre el desarrollo de las asignaturas.
Participe en el diagnóstico integral y permanente de la docencia en la universidad.


Los resultados serán analizados por los docentes y cuerpos colegiados con el fin de generar planes de mejoramiento para un mejor desarrollo de los cursos.



Cómo entrar

Login : Código de Estudiante [guión]
Código del programa Académico
Ejemplo: Si su programa académico es el 3743 y su código de estudiante es 9910177, entonces su login es: 9910177-3743
Contraseña : Número de Referencia (ver parte inferior del recibo de pago). Para estudiantes de POSTGRADO la contraseña es el número del documento de identidad.

Ingresar al sistema




▶ Login

▶ Contraseña

! No guarde su contraseña en el navegador web.
Recuerde que ésta es personal e intransferible.

Ingresar



Listo

<http://evaluacioncursos.univalle.edu.co>