كلية الهندسة

قسم الهندسة المعلوماتية

مقرر خوارزميات بحث ذكية

د. غزوان علي ريّا

محاضرات الأسبوع الرابع

الفصل الأول 2024–2025

تتمة خوارزميات البحث من دون معلومات

خوارزمية البحث ذو التكلفة الموحدة.

خوارزمية البحث ثنائي الاتجاه.

سنكمل بخوارزميات البحث من دون معلومات ونذكر آخر خوارزميتين خوارزمية البحث ذو التكلفة الموحدة وخوارزمية البحث ثنائي الاتجاه.

البحث ذو التكلفة الموحدة Uniform cost search UCS:

وهو شبيه بخوارزمية ديجكسترا، يمكن تعريف خوارزمية البحث ذو الكلفة الموحدة بأنها خوارزمية بحث عمياء تستعمل أقل كلفة تجميعية لتجد طريق من المصدر للهدف.

أي أنها خوارزمية تجد الطريق الأقل تكلفة من المصدر للهدف،

يتم توسيع العقد فيها بدءاً من الجذر حسب التكلفة التجميعية الأقل، ويتم تحقيق هذه الخوارزمية باستخدام رتل الأولوية Priority Queue.

كما ذكرنا سابقاً في حل كانت كل الكلف متساوية يعطي البحث بالعرض أولاً BFS حلاً أفضلي لأنه يقوم دوماً بتوسعة العقدة الأكثر سطحية والتي لم يتم توسعتها مسبقاً.

فبتعديل بسيط على BFS نستطيع إيجاد خوارزمية تعطي حل أفضلي عند عدم تساوي الكلف ويكون هذا بالشكل الآتي:

فبدلاً من توسعة العقدة الأكثر سطحية، يقوم البحث ذو التكلفة الموحدة بتوسيع العقدة n والتي تملك أقل كلفة طريق g(n)،

يتم هذا عن طريق تخزين المجموعة الأمامية frontier **كرتل أولوية** مرتب حسب الكلف g(n).

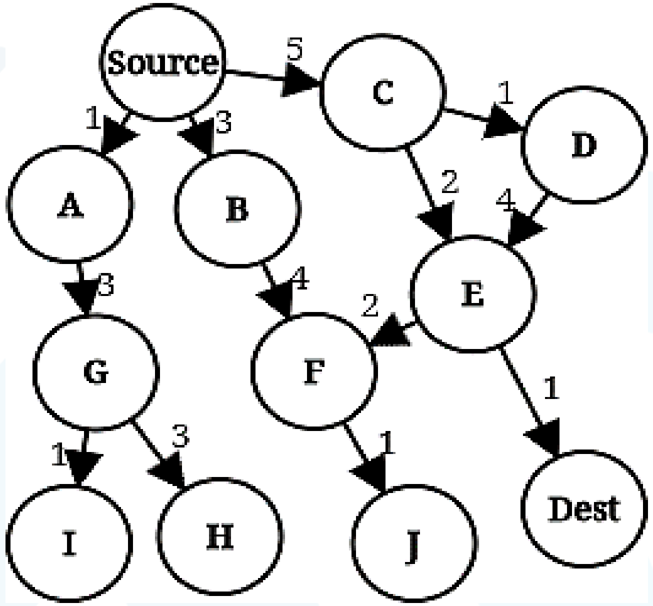
خوارزمية البحث ذو التكلفة الموحدة هي كما يلي:

```
Insert the root node into the priority
queue
Repeat while the queue is not empty
Remove the element with the highest
priority
If the removed node is the destination
Print total cost and stop the algorithm
Else
Enqueue all the children of the current
node to the priority queue with their
cumulative cost from the root as priority
```

سنقوم بشرح خطوات الخوارزمية عبر تمرينين:

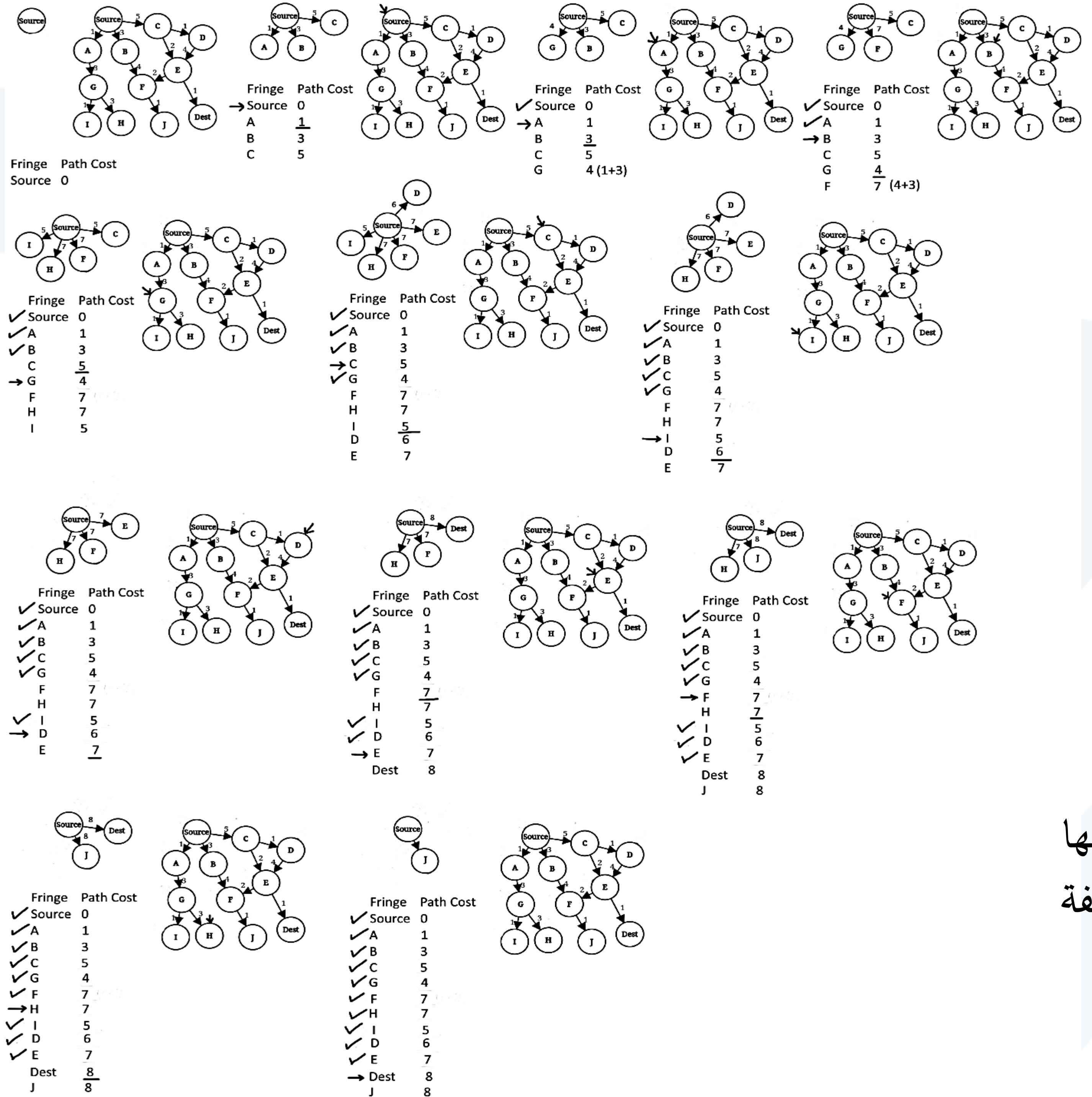
تمرين:

طبق خوارزمية البحث ذو التكلفة الموحدة USC على البيان التالي انطلاقاً من العقدة Source حتى العقدة Dest أي المطلوب هو إيجاد أقل طريق كلفة بين العقدتين.



الحل:

سنقوم باستعمال المجموعة الهامشية fringe حيث نضع بها العقد أبناء العقدة التي نتعامل معها وسنربطها مع تكلفة الطريق path cost الذي يؤدي للعقدة في جدول كما يلي.



Fringe	Path Cost
Source	0

Fringe	Path Cost
→ Source	0
A	1
B	3
C	5

Fringe	Path Cost
✓ Source	0
→ A	1
B	3
C	5
G	4 (1+3)

Fringe	Path Cost
✓ Source	0
✓ A	1
→ B	3
C	5
G	4
F	7 (4+3)

Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
C	5
→ G	4
F	7
H	7
I	5

Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
→ C	5
✓ G	4
F	7
H	7
I	5
D	6
E	7

Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
F	7
H	7
→ I	5
D	6
E	7

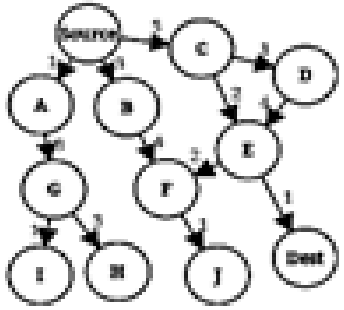
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
F	7
H	7
✓ I	5
→ D	6
E	7

Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
F	7
H	7
✓ I	5
✓ D	6
→ E	7
Dest	8

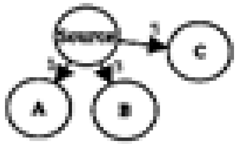
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
→ F	7
H	7
✓ I	5
✓ D	6
✓ E	7
Dest	8
J	8

Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
✓ F	7
→ H	7
✓ I	5
✓ D	6
✓ E	7
Dest	8
J	8

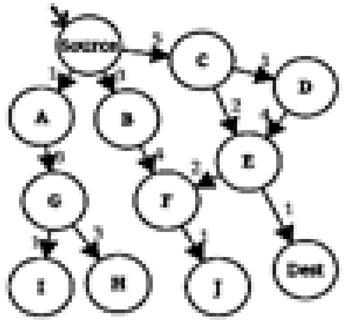
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
✓ F	7
✓ H	7
✓ I	5
✓ D	6
✓ E	7
→ Dest	8
J	8

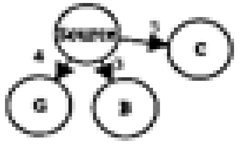


Fringe	Path Cost
Source	0

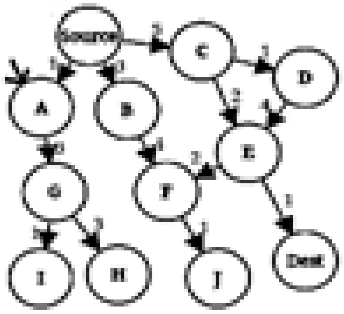


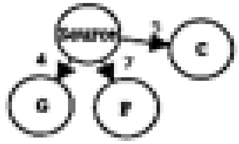
Fringe	Path Cost
→ Source	0
A	1
B	3
C	5



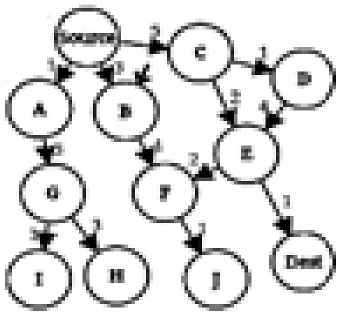


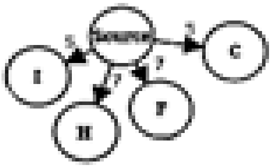
Fringe	Path Cost
✓ Source	0
→ A	1
B	3
C	5
G	4 (1+3)



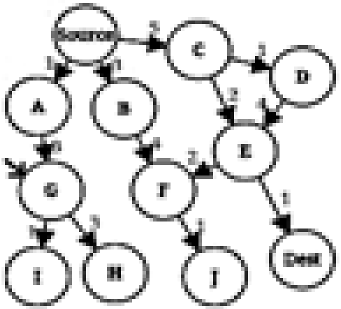


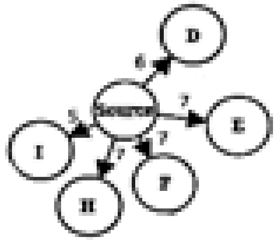
Fringe	Path Cost
✓ Source	0
✓ A	1
→ B	3
C	5
G	4
F	7 (4+3)



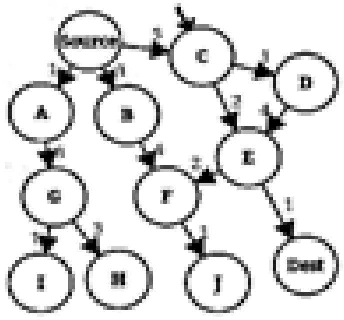


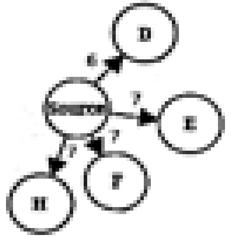
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
C	5
→ G	4
F	7
H	7
I	5



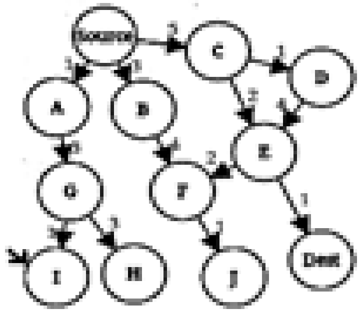


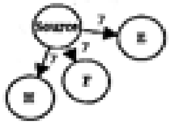
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
→ C	5
✓ G	4
F	7
H	7
I	5
D	6
E	7



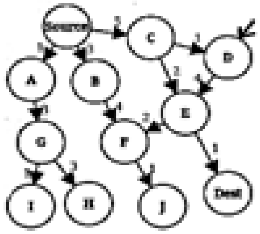


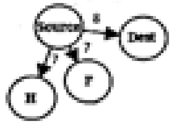
Fringe	Path Cost
✓ Source	0
✓ A	1
✓ B	3
✓ C	5
✓ G	4
F	7
H	7
→ I	5
D	6
E	7



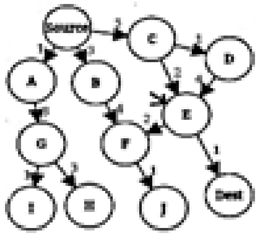


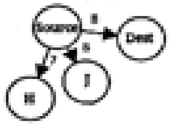
	Fringe	Path Cost
✓	Source	0
✓	A	1
✓	B	3
✓	C	5
✓	G	4
	F	7
	H	7
✓	I	5
→	D	6
	E	7



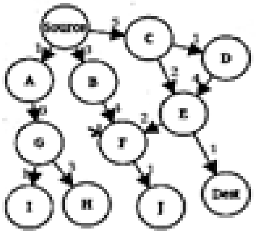


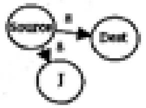
	Fringe	Path Cost
✓	Source	0
✓	A	1
✓	B	3
✓	C	5
✓	G	4
	F	7
	H	7
✓	I	5
✓	D	6
→	E	7
	Dest	8



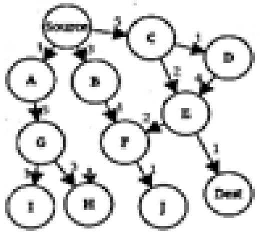


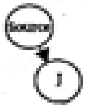
	Fringe	Path Cost
✓	Source	0
✓	A	1
✓	B	3
✓	C	5
✓	G	4
→	F	7
	H	7
✓	I	5
✓	D	6
✓	E	7
	Dest	8
	J	8



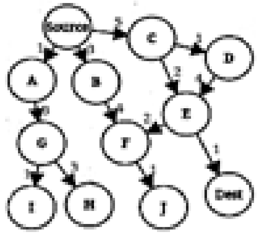


	Fringe	Path Cost
✓	Source	0
✓	A	1
✓	B	3
✓	C	5
✓	G	4
✓	F	7
→	H	7
✓	I	5
✓	D	6
✓	E	7
	Dest	8
	J	8





	Fringe	Path Cost
✓	Source	0
✓	A	1
✓	B	3
✓	C	5
✓	G	4
✓	F	7
✓	H	7
✓	I	5
✓	D	6
✓	E	7
→	Dest	8
	J	8



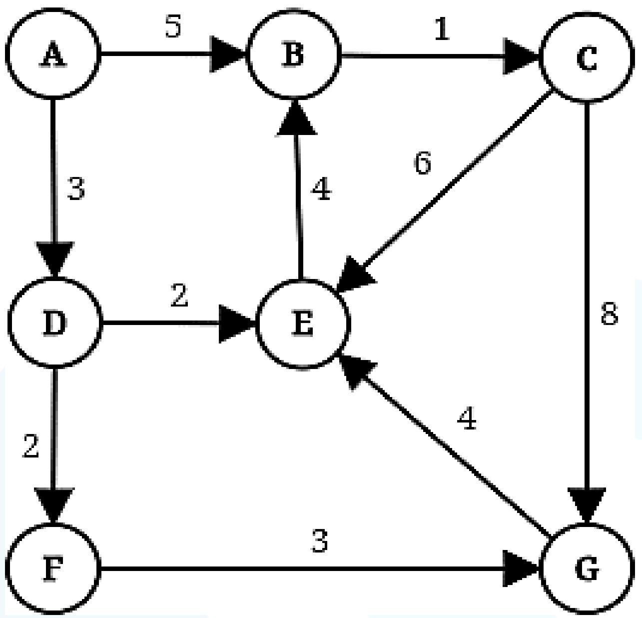
- في بداية الخوارزمية يكون هناك عنصر واحد في المجموعة الهامشية هو عقدة البداية (في هذا المثال هي source وكلفة الطريق 0 ثم نضيف جميع أبناء العقدة مع كلف طريقها فنضيف على الجدول 5 C 3 B 1 A
- نختار العقدة ذات الكلفة الأقل ونضيف أبناءها فنضيف العقدة G وتكون كلفة الطريق بما أننا اخترنا العقدة A هي كلفة الطريق إلى A ثم كلفة الطريق إلى G أي 4،
- ثم ننتقل لأقل عقدة تكلفةً وهي B ونضيف أبناءها F فتكون كلفة الطريق هي كلفة الطريق إلى B ثم كلفة الطريق إلى G وهي 7 ونكمل بنفس الطريقة،
- عند وجود أكثر من عقدة لهما أقل تكلفة (كما في الحالة 6 يمكن اختيار C أو I حيث أن لهما نفس أقل تكلفة 5 بين كل العقد)
- عندما نصل لعقدة من دون أبناء لا نقوم بشيء وننتقل للعقدة التالية في الرتل (ترتيب العقد من أقل تكلفة حتى أعلى تكلفة)
- عندما نصل للعقدة E نلاحظ أن أحد أبنائها F فنقارن قيمة F في الرتل والتي هي 7 مع قيمة الطريق من E إلى F وهي 7 $9 = 2 + 7$ فنلاحظ أن 7 أصغر من 9 فلا نغير من قيمة F (ولكن لو كانت قيمة الطريق من F إلى E مثلاً 5 كنا استبدلنا قيمة F في الرتل) ونكمل،
- نلاحظ في الحالة 9 أننا أضفنا عقدة الهدف للرتل، ولكن سنستمر بالخوارزمية حتى تصبح عقدة الهدف dest هي العقدة الحالية في الرتل وعندما تخرج من الرتل نكون حصلنا على أقصر طريق وهو source C E dest بكلفة $8 = 1 + 2 + 5$.

تمرين 2:

قم بإيجاد أقصر طريق انطلاقاً من العقدة A حتى العقدة الهدف G في البيان التالي،

سنقوم هنا بذكر المجموعة الهامشية وقيمة الحل.

Fringe	Path cost
[A]	0
[A,D]	3
[A,B]	5
[A,D,E]	2+3=5
[A,D,F]	2+3=5
[A,B,C]	5+1=6
[A,D,F,G]	5+3=8



في المرحلة قبل الأخيرة عندما نختبر العقدة C نلاحظ أن الطريق إلى G كلفته 14 = 8 + 6 وبالتالي نحافظ على قيمة G في الرتل كما هي حيث أن كلفة الوصول الحالية إليها أقل، وبعدها ننتقل إلى العقدة الهدف وبما أن العقدة G كلفة الطريق للعقدة E هي أكبر من كلفة الطريق في الرتل فنحافظ على قيمة E في الرتل وبالتالي نكون انتهينا من العقدة الهدف وأخرجناها من الرتل فيكون الحل الأفضلي هو A D F G بكلفة 8.

ملاحظة:

تنتهي خوارزمية ال UCS عندما نزيل العقدة الهدف من الرتل وليس عندما نقوم بتوسعة العقد الهدف.

أي أنه في خوارزميتي ال A* وال UCS يتم اختبار الهدف لتحديد إن كانت الحالة حالة نهائية أم لا عندما تتم إزالة العقدة من الرتل وهذا يضمن لنا أننا لن نجد طريق قصير ومكلف قبل طريق طويل ورخيص.

البحث ثنائي الاتجاه Bidirectional Search

- هو عبارة عن خوارزمية للبحث في البيان تبحث عن أقصر مسار من عقدة البداية لعقدة النهاية،

وفي بيان موجه تقوم هذه الخوارزمية ببحثين متزامنين: واحد للأمام من الحالة الأولية والآخر للخلف من الهدف،

تتوقف الخوارزمية عن التقاء البحثين.

- نستطيع استعمال البحث ثنائي الاتجاه في الحالات التالية: عندما تكون عقدة البداية والهدف مميزتان ومعرفتان بشكل واضح (أي أن هناك عقدة هدف مذكورة بشكل صريح)، وعندما معامل التفرع هو نفسه في الاتجاه الأمامي والخلفي.

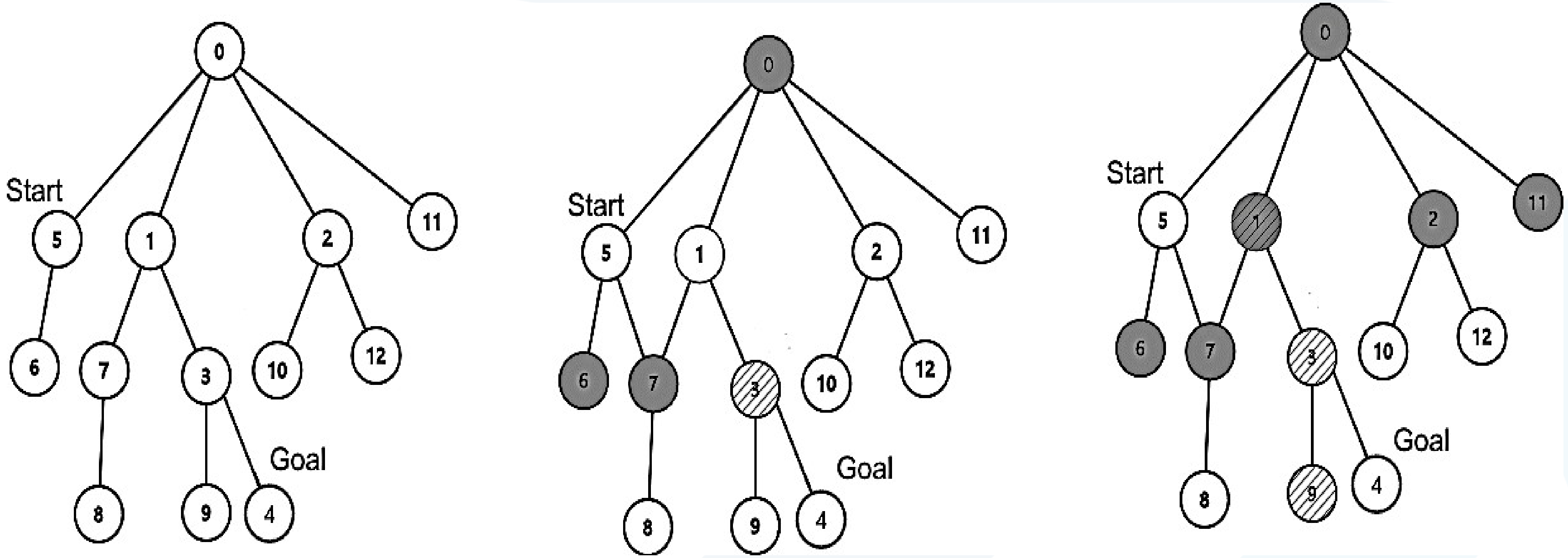
خوارزمية البحث ثنائي الاتجاه هي كما يلي:

```
startq = Queue for BFS from start node
endq = Queue for BFS from end node
parent = Array where startparent[i] is parent of node i
visited = Array where visited[i] = True if node i has been
encountered
while startq is not empty and endq is not empty
perform next iteration of BFS for startq (also save the
parent of the children in the parent array)
perform next iteration of BFS for endq
if we have encountered the intersection node save the
intersection node
break
using intersection node find the path using parent array
```

وسنقوم بشرح الخوارزمية من خلال مثال بسيط.

تمرين 1:

ليكن لدينا البيان الموضح بالشكل. طبق خوارزمية البحث ثنائي الاتجاه عليه علماً أن عقدة البداية هي 5 وعقدة النهاية هي 4.



يوضح الشكل العملية حيث أن العقد المضافة في الجهة الأمامية (جهة عقدة البداية) ملونة باللون الرمادي، والعقد المضافة في الجهة الخلفية (جهة عقدة النهاية) مُشار لها بالخطوط

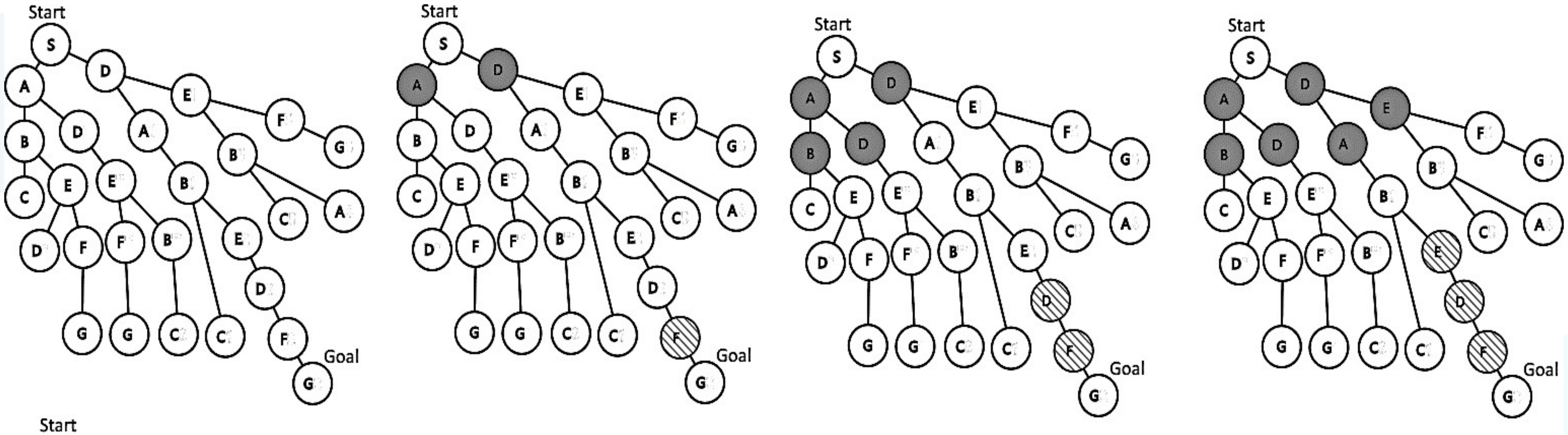
في الخطوة الثالثة نلاحظ أن العقدة 1 تم اكتشافها من الجهتين وأصبحت عقدة تقاطع وتتوقف الخوارزمية عند اكتشاف عقدة تقاطع وترد إما 4 3 1 7 5 أو 4 3 1 0 5

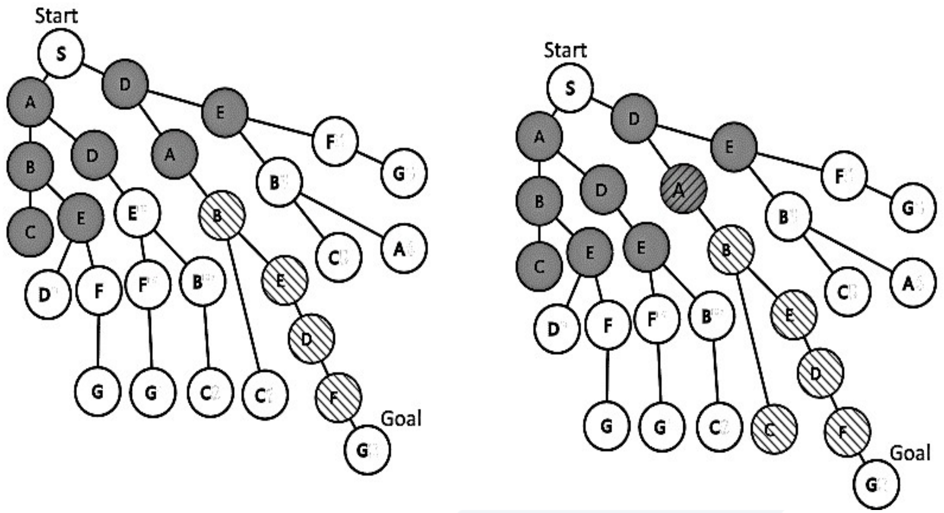
عند تطبيق الخوارزمية سنقوم بإنشاء جدولين جدول للجهة الأمامية وجدول للجهة الخلفية ونقوم بنفس خطوات خوارزمية BFS في الاتجاهين.

Forward		Backward	
Open	Closed	Open	Closed
5	-	4	-
0 7 6	5	3	4
7 6 1 2 11	0 5	1	3 4

يمكن إرشاد البحث ثنائي الاتجاه عبر التقدير الحدسي heuristic estimate للمسافة من عقدة البداية لعقدة الهدف لإيجاد أقصر طريق ممكن.

تمرين 2: طبق خوارزمية البحث ثنائي الاتجاه Bidirectional Search على البيان التالي:





Forward		Backward	
Open	Closed	Open	Closed
S	-	G	-
A D	S	F	G
D B D	A S	D	F G
B D A E	D A S	E	D F G
D A E C E	B D A S	B	E D F G
A E C E E	D B D A S	C A	B E D F G

قمنا في التمرين السابق بإنشاء مجموعتين Open, Closed لجهتي البحث في البيان وتابعنا حتى الوصول لعقدة تقاطع وحينها تتوقف الخوارزمية.

مقاييس أداء البحث ثنائي الاتجاه:

البحث ثنائي الاتجاه تام في حال كانت طريقة البحث في كلا الاتجاهين .BFS.

يعتبر أفضلياً optimal في حال استعمال BFS بشرط أن تكون الوصلات تملك كلف متساوية، التعقيد الزمني والمساحي هو $O(\frac{bd}{2})$.

يمثل الجدول الآتي مقارنة بين خوارزميات البحث بدون معلومات Uninformed Search.

	BFS	UCS	DFS	DLS	IDDFS	Bidirectional
التمام	تام[أ]	تام[أ ب]	لا	لا	تام[أ]	تام[أ ج]
الزمن	$O(b^d)$	$O(b^{1+\frac{c^*}{\varepsilon}})$	$O(b^m)$	$O(b^l)$	$O(b^d)$	$O(b^{\frac{d}{2}})$
الفضاء	$O(b^d)$	$O(b^{1+\frac{c^*}{\varepsilon}})$	$O(bm)$	$O(bl)$	$O(bd)$	$O(b^{\frac{d}{2}})$
الأفضلية	أفضلي[د]	أفضلي	لا	لا	أفضلي[د]	أفضلي[د ج]

حيث:

b: معامل التفرع لشجرة البحث.

d: هو عمق أقل حل مكلف.

m: هو عمق الشجرة الأعظمي وقد يكون العمق غير منته.

l: هو العمق المحدود (العمق الذي لا يمكن تجاوزه).

أ: تكون الخوارزمية تامة في حال كان معامل التفرع منته.

ب: تكون الخوارزمية تامة إن كانت كل الكلف موجبة.

ج: في حال طبقنا BFS في الاتجاهين.

د: يعطي حلاً أفضلياً عندما تكون كل الكلف متساوية.