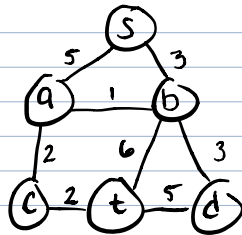


1. Search



n	h(n)
s	5
a	4
b	3
c	1
d	4
t	0

Breadth-First Search

node	came from	visited
s		✓ 1
a	s	✓ 2
b	s	✓ 3
c	a	✓ 4
d	b	✓ 5
t	b	✓ 6

Queue:

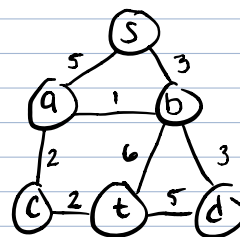
~~s~~
~~a~~
~~b~~
~~c~~
~~d~~
~~t~~

Order: S, A, B, C, D, T

1. S - visited, enqueue
2. a & b visited, dequeue s
3. make "a" working node, "b" already visited, "c" is now visited: enqueue
4. make "b" working node, d & t visited, enqueue them
5. queue is complete with nothing else to add
6. make "c" working node, its relations have already been visited, dequeue
7. make "d" working node, it's relations have been visited, dequeue
8. make "t" working node, it's the target, dequeue. Empty.

Depth-First Search

node	came from	visited
s		✓ 1
a	s	✓ 2
b	a	✓ 3
c	t	
d	b	✓ 4
t	d	✓ 5



Stack:

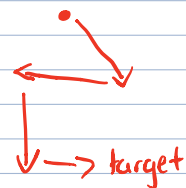
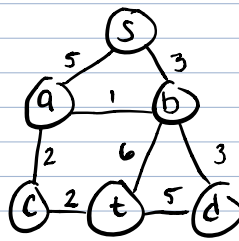
c
t
d
b
a
s

Order: S, A, B, D, T

1. visit s, push
2. visit a, push
3. visit b, push
4. visit d, push
5. visit t, target! push.
6. visit c, push
7. pop c, t, d, b, a, s, empty stack.

Uniform-Cost Search

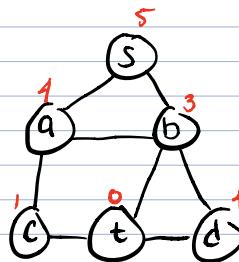
node	came from	$g(n)$	visited?
s	-	0	✓
a	s b	5 4	✓
b	s	3	✓
c	a	6	✓
d	b	6	
t	b c	7 8	✓



Order: s, b, a, c, (t)

Greedy Best First Search

node	came from	$h(n)$	visited?
s	-	5	✓
a	-	1	
b	s	3	✓
c	-	1	
d	-	4	
t	b	0	✓

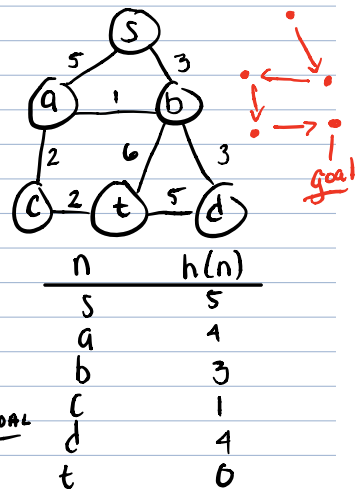


n	$h(n)$
s	5
a	1
b	3
c	1
d	4
t	0

Order: s, b, (t) - target!

A*

node	came from	$g(n)$	$h(n)$	$f(n)$	visited
s	-	0	5	$0+5=5$	✓ 1
a	s	5	1	$5+1=6$	
b	s	3	3	$3+3=6$	✓ 2
a	b	$3+1$	1	$4+1=5$	✓ 3
d	b	$3+3$	4	$6+4=10$	
t	b	$3+6$	0	$9+0=9$	
c	a	$3+1+2$	1	$6+1=7$	✓ 4
t	c	$3+1+2+2$	0	$8+0=8$	✓ 5



Order: s, b, a, c, (t) - goal!