

AI

Part C

3.1

1. ~~Dep~~ False. Depth-first search doesn't expand until it reaches a leaf node.

~~2. ~~False~~ $h(n) = 0$ seems~~

2. False. $h(n) = 0$ seems "optimal" however except the 8-puzzle has everything correct to the goal state, $h(n) = 0$ doesn't exist.

Also, what kind of method is used for this $h(n)$ is not mentioned.

3. ~~A~~ False, it was used on Shakey the Robot.

Also, it is ~~an~~ a heuristic algorithm, and ~~it is a kind of level set~~ It is also wrong because computers only take discrete (digital) values.

4. True. Breadth-first search is complete period. But one should make sure there is no cyclic structure.

5. True. ~~Man~~ Manhattan distance is an admissible heuristic because rooks only move vertically or ~~right~~ horizontally. Therefore, Manhattan distance should be the most optimal for a rook to move from A to B.

3.2

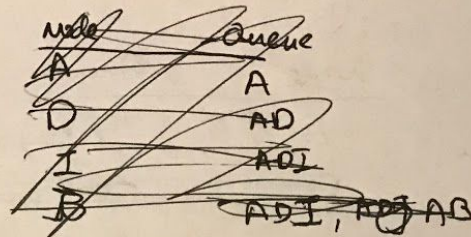
1)	Node	Queue	to
	A	AA (A)	
(not optimal)	B	AB	
	C	AB, AC	
	D	AB, AC, AD	
	E	ABE, AC, AD	
	F	ABE, ABF, AC, AD	
	G	ABE, ABF, ACG, AD	
	H	ABE, ABF, ACG, ACH, AD	
	I	ABE, ABF, ACG, ACH, ADI	
	J	ABE, ABF, ACG, ACH, ADI, ADJ	
	K	ABEK, ABF, ACG, ACH, ADI, ADJ	
	L	ABEL ABEL, ABFL done	goal state reached.

2) Uniform Cost

(optimal)	Node	Queue
	A	A
	D	AD
	B	ADI (cost) AD, AB AB
	C	AD, AB, AC
	IAD	ADI, AB, AC
	J	ADI, ADIJ, AB, AC
	F	ADI, ADIJ, ABF, AC
flip order	E	ADI, ADIJ, ABF, ABE, AC
	G	ADI, ADIJ, ABF, ABE, ACG
	H	ADI, ADIJ, ABF, ABE, ACG, ACH
	Q	ADI, ADIJQ, <u>ACGN</u>
	N	ADI, ADIJQ, ACGN

least cost goal state

c) greedy



Node	Queue
A	A
B	AB (heuristic)
F	ABF
L	<u>ABFL</u> (not optimal)

d) A*

(optimal)

Node	Queue
A	A(4)
B, C, D	AB(6+2=8), AC(6+3=9), AD(8)
E, F, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z	ABE(12), ABFL(11), AC(9), AD(8)
I, J	ABE(12), ABFL(11), AC(9), ADI(16), ADJ(11)
G, H	ABE(12), ABFL(11), ACG(11), ADI(16), ADJ(11), ACH(15), ...
	alphabetical
L, M	ABE(12), ABFL(14), ABFM(17), ACG(11) ...
N, O	ABE(12), ABFL(14), ABFM(17), ACGN(13) ^{ACG(14)} ...
Q	ADJQ(17)
K	ABEK(18), ABFL(14), ABFM(17), ACGN(13), ACGO(14), ADIL(16), ADJQ(17)
Till now, nobody is shorter than the known viable path <u>ACGN(13)</u> .	

e) hill climbing

NOT
HEURISTIC!

node	Queue
A	A(0) A(0)
B	AB(8)
D	AD(1)
I	ADI(2)

↳ stuck on local maximum.

f) local beam

Node	Queue
A AND H	A, H
D AND P	AD, HP
I AND Q done	ADI, HP (local maximum)

Both ends up landing local maxima.

2. From previous examples,
uniform cost, A*

are able to find optimal solutions.

3.3

1. I will need:

- word length
 - ~~two~~ All the letters in a word that also occurred in other words.
 - The ~~recurring~~ location of those recurring letters in each word.
 - Shouldn't be shorter than 3 letters
Shouldn't be longer than 6 letters.
- Words
- a ~~structure~~ contains the space with length marked
 - a mapping structure that remembers where the row and column share space / letter. Ex. (1D(3), 2D(2))
- map

