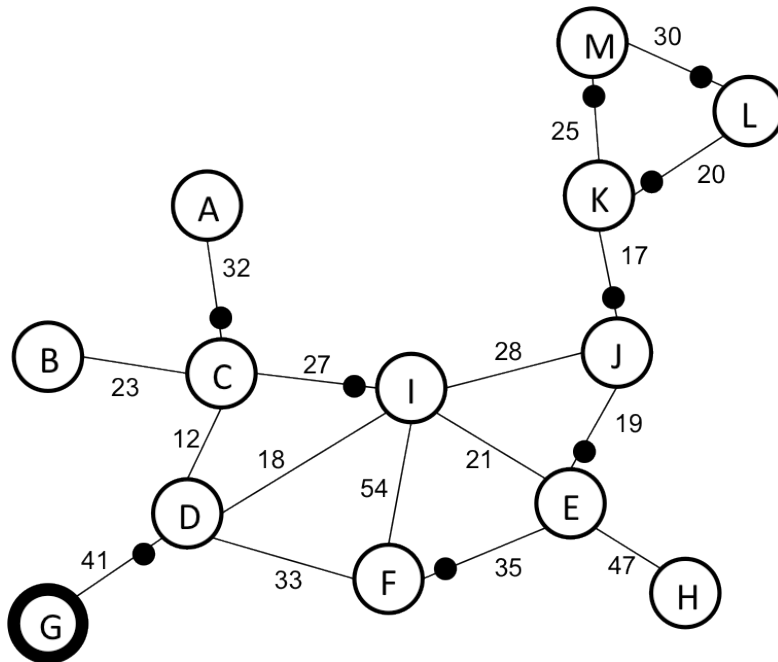


## CS 3600 – Introduction to AI

### Practice Search Problem

Consider the search space diagrammed below. Action costs are given next to each arc. Use “E” as the initial state and “G” as the goal state. The successor function for a node starts with the arc with the black dot next to it first, and then generates successors in a counter-clockwise fashion. For example,  $\text{Successors}(E) = \{J, I, F, H\}$  and  $\text{Successors}(J) = \{K, I, E\}$ .



1. What order will nodes be visited using the **breadth-first** algorithm? Ignore action costs. List the visit order and the final solution.
2. What order will nodes be visited using the **depth-first** algorithm? Ignore action costs. List the visit order and the final solution.
3. What order will nodes be visited using the **uniform cost search** algorithm? List the visit order and the final solution.
4. What order will nodes be visited using the best-first algorithm? Use the heuristic function given below. List the visit order and the final solution.

$h(A) = 65$	$h(D) = 41$	$h(H) = 100$	$h(K) = 50$
$h(B) = 25$	$h(E) = 85$	$h(I) = 55$	$h(L) = 100$
$h(C) = 50$	$h(F) = 30$	$h(J) = 110$	$h(M) = 95$

1. What order will nodes be visited using the **breadth-first** algorithm? Ignore action costs. List the visit order and the final solution.

Open: E  
Closed: nil

Visit E

Open: J I F H  
Closed: E

Visit J

Open: I F H K I E  
Closed: E J

\* Note: I am placing duplicates on open, but ignore later

Visit I

Open: F H K I E C D F E J  
Closed: E J I

Visit F

Open: H K I E C D F E J D  
Closed: E J I F

Visit H

Open: K I E C D F E J D  
Closed: E J I F H

Visit K

Open: I E C D F E J D L M  
Closed: E J I F H K

Visit C

\* Note: Skipping I and E, which are already visited

Open: D F E J D L M A B D  
Closed: E J I F H K C

Visit D

Open: F E J D L M A B D G F  
Closed: E J I F H K C D

\* Can jump straight to G if cost of goal() isn't great.

Visit L

\* Skip F, E, J, D, all of which are already visited

Open: M A B D G F M  
Closed: E J I F H K C D L

Visit M

Open: A B D G F M  
Closed: E J I F H K C D L M

Visit A

Open: B D G F M  
Closed: E J I F H K C D L M A

Visit B

Open: D G F M  
Closed: E J I F H K C D L M A B

Visit G \* Skip D ,which is already visited

Open: F M  
Closed: E J I F H K C D L M A B G

**Visit order:** E J I F H K C D L M A B G

**Solution:** E->I, I->D, G->D

2. What order will nodes be visited using the **depth-first** algorithm? Ignore action costs. List the visit order and the final solution.

Open: E  
Closed: nil

Visit E

Open: J I F H  
Closed: E

Visit J

Open: K I I F H  
Closed: E J

Visit K

Open: L M I I F H  
Closed: E J K

Visit L

Open: M M I I F H  
Closed: E J K L

Visit M

Open: M I I F H  
Closed: E J K L M

Visit I \* Skip M, already visited

Open: C D F I F H  
Closed: E J K L M I

Visit C

Open: A B D D F I F H  
Closed: E J K L M I C

Visit A

Open: B D D F I F H  
Closed: E J K L M I C A

Visit B

Open: D D F I F H  
Closed: E J K L M I C A B

Visit D

Open: G F D F I F H  
Closed: E J K L M I C A B D

Visit G

Open: F D F I F H  
Closed: E J K L M I C A B D G

**Visit order:** E J K L M I C A B D G

**Solution:** E->I, I->C, C->D, D->G

3. What order will nodes be visited using the **uniform cost search** algorithm? List the visit order and the final solution.

Open: E(0)

Closed: nil

Visit E

Open: J(19) I(21) F(35) H(47)

Closed: E

Visit J

Open: I(21) F(35) K(36) H(47)

Closed: E J

Visit I

Open: F(35) K(36) D(39) H(47) C(48)

Closed: E J I

Visit F

Open: K(36) D(39) H(47) C(48)

Closed: E J I F

Visit K

Open: D(39) H(47) C(48) L(56) M(61)

Closed: E J I F K

Visit D

Open: H(47) C(48) L(56) M(61) G(80)

Closed: E J I F K D

Visit H

Open: C(48) L(56) M(61) G(80)

Closed: E J I F K D H

Visit C

Open: L(56) M(61) B(71) G(80) A(80)

Closed: E J I F K D H C

Visit L

Open: M(61) B(71) G(80) A(80)

Closed: E J I F K D H C L

Visit M

Open: B(71) G(80) A(80)

Closed: E J I F K D H C L M

Visit B

Open: G(80) A(80)

Closed: E J I F K D H C L M B

Visit G

Open: A(80)

Closed: E J I F K D H C L M B G

**Visit order:** E J I F K D H C L M B G

**Solution:** E->I, I->D, D->G

4. What order will nodes be visited using the best-first algorithm? Use the heuristic function given below. List the visit order and the final solution.

Open: E(85)

Closed: nil

Visit E

Open: F(65) I(76) J(129) H(147)

Closed: E

Visit F

Open: I(76) D(109) J(129) H(147)

Closed: E F

Visit I

Open: D(80) C(98) J(129) H(147)

Closed: E F I

Visit D

Open: G(80) C(98) J(129) H(147) G(80)

Closed: E F I D

Visit G

Open: C(98) J(129) H(147) G(80)

Closed: E F I D G

**Visit order:** E F I D G

**Solution:** E->I, I->D, D->G